

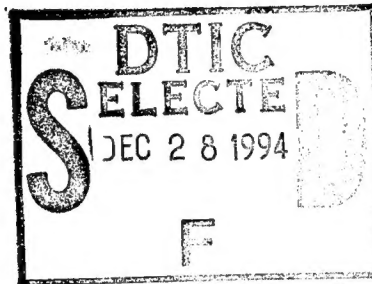


NRL/MR/5360--94-7618

## Data Input for RADARPD: Window Version

MOLLY M. YEN

*Airborne Radar Branch  
Radar Division*



December 13, 1994

Approved for public release; distribution unlimited.

19941223 063

# REPORT DOCUMENTATION PAGE

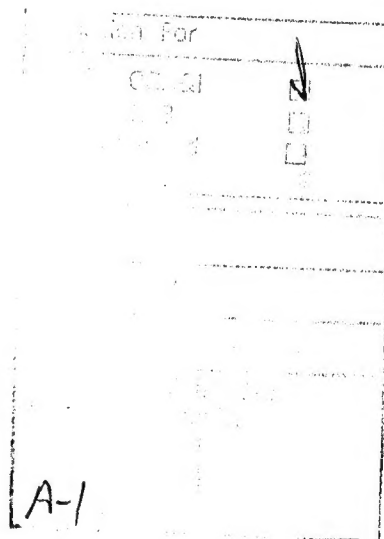
Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)		2. REPORT DATE  December 13, 1994	3. REPORT TYPE AND DATES COVERED	
4. TITLE AND SUBTITLE  Data Input for RADARPD: Window Version			5. FUNDING NUMBERS	
6. AUTHOR(S)  Molly M. Yen				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  Naval Research Laboratory Washington, DC 20375-5320			8. PERFORMING ORGANIZATION REPORT NUMBER  NRL/MR/5360-94-7618	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)  Naval Air Warfare Center Ad Patuxent River, MD 20670-5304			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT  Approved for public release; distribution unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)  An X Window interface inputting data to the RADARPD computer program has been developed. The RADARPD program reads in an input data file and calculates the probability of detection. A brief introduction contrasts the previous VAX based text editor input interface with the new SUN based X Window input interface. After the introduction is a detailed description of the window interface that explains the organization and format of the input window; following that is step-by-step instructions on how to enter and edit data files using the new interface. Concluding the report is the source code.				
14. SUBJECT TERMS			15. NUMBER OF PAGES  58	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT  UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE  UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT  UNCLASSIFIED	20. LIMITATION OF ABSTRACT  UL	

## CONTENTS

I. INTRODUCTION.....	1
II. WINDOW INTERFACE FORMAT.....	1
III. OPERATION PROCEDURE.....	2
IV. ACKNOWLEDGMENT.....	2
Appendix A.....	A-1
Appendix B.....	B-1



## DATA INPUT FOR RADARPD: WINDOW VERSION

### I. INTRODUCTION

RADARPD is a computer program which calculates the probability of detection. The program reads in a user inputted data file and then performs the computation. Recently it has been converted from the VAX 11-750 VMS operating system to the SUN Solaris operating system. Previously on the VAX computer, the user had to use a text editor to create or edit the input data file. This input procedure was quite tedious since the data file usually consisted of a long string of numbers, simply separated by commas (Figure 1 ). The user not only had to remember the order in which to input the data values, but had to remember what each data value represented as well. A window interface for creating and editing the input data file has been developed for the SUN computer, using C language and the NRLx Toolkit, an X Window toolkit. The window interface uses graphical objects such as buttons and labeled text fields. It has not only simplified the input process, but is extremely user friendly as well.

### II. WINDOW INTERFACE FORMAT

The window interface consists of a main menu window, Figure 2, and nineteen subwindows, Figure 3 through Figure 21. Each subwindow contains a group of related data values, such as Signal Parameters, Antenna Characteristics, etc. The user is able to open a subwindow or perform certain functions, such as saving the data file and executing the calculation, by simply clicking on a button from the main menu window. Different types of data can be entered, such as integers, floating points, and character strings. There are ten fields on each subwindow, four representing integer values (Item1, Item2, Item3, and Item4), and six representing floating point values (Var1, Var2, Var3, Var4, Var5, and Var6). Each field is labeled with its name and its unit (deg., dB, etc.), if applicable. Not all of the ten fields are used for each subwindow function. For example, only six fields are used for Signal Parameters, Figure 3, while all the fields are used for Antenna characteristics, Figure 8. Only one subwindow, Title for Printout and Plot, Figure 12, contains character string fields.

### III. OPERATION PROCEDURE

The user can use this window interface program to create a new data file or edit an existing data file. Since each case has different requirements, the data file does not have a fixed size. The user can click and enter data for as many subwindows as needed. The Execute subwindow, Figure 21, is the only required subwindow for all cases, and should always be the last subwindow to be entered.

The input procedure is as follows:

- (1) Create a new data file:
  - (a) Click on "Open a new data file" button.
  - (b) Click the desired subwindow button on the main menu.
  - (c) Enter the values into the text fields.
  - (d) Save the window when completed by clicking on "Save this new window" button.
  - (e) Repeat b, c, and d as many times as needed.
  - (f) Close the new data file by clicking on the "Close the new data file" button.
- (2) Retrieve and edit an existing data file:
  - (a) Click on "Open an existing data file" button.
  - (b) Click on the desired subwindow button that needs to be edited.
  - (c) The old values will appear in the subwindow, Figure 10. Make the proper changes.
  - (d) Save the retrieved window when completed by clicking on the "Save this in window" button.
  - (e) Repeat b, c, and d as many times as needed.
  - (f) Close the retrieved data file by clicking on the "Close this existing data file" button.

### IV. ACKNOWLEDGMENT

I would like to thank Mr. Fred Staudaher who gave me the idea to upgrade the RADARPD input data file to a window interface version.

Appendix A  
LIST OF FIGURES

Figure 01	VAX Input Data File.....	A-02
Figure 02	Main Menu .....	A-03
Figure 03	Signal Parameters.....	A-04
Figure 04	Radar Detection Parameters.....	A-05
Figure 05	AEW Geometry.....	A-06
Figure 06	Target Height Limits.....	A-07
Figure 07	Target Range Limits.....	A-08
Figure 08	Antenna Characteristics.....	A-09
Figure 09	Antenna Data Points.....	A-10
Figure 10	Clutter Characteristics.....	A-11
Figure 11	Cancellation Characteristics.....	A-12
Figure 12	Title for Printout & Plot.....	A-13
Figure 13	Target Velocity Limits.....	A-14
Figure 14	PD Contour Plot Data.....	A-15
Figure 15	Extended Surface Target Parameters.....	A-16
Figure 16	Parameters for Isometric Plot.....	A-17
Figure 17	Antenna Pattern Print/Plot Parameters.....	A-18
Figure 18	Atmospheric Attenuation Parameters.....	A-19
Figure 19	Rain Attenuation Parameters.....	A-20
Figure 20	Jammer Parameters.....	A-21
Figure 21	Execution.....	A-22

```

2,,3000,,,1.,5.,5.66,732.6,20.,
3,1,1,4,2,,229.,6.4,30.,,
4,,,,,25000.,,,,,,
5,,,,,10.,50.,2.,,,
6,,,,,20.,200.,2.,,,
7,,,,,3.,34.,18.,,,45.
9,1,,,1,-43.8,0.8,0.2,10.,,
112,,,,,,
Test Run
May 30, 1994
1623,8,,,40.,0.2,5.,,,
1,,,,,10.,,,,,

```

Figure 1 Old VAX Input Data File

Input Data for RADARPD	
RADARPD Input Menu	
Signal Parameters	
Radar Detection Parameters	
AEW Geometry	
Target Height Limits	
Target Range Limits	
Antenna Characteristics	
Antenna Data Points	
Clutter Characteristics	
Cancellation Characteristics	
Title for printout & plot	
Target Velocity Limits	
PD Contour Plot Data	
Extended Surface Target Parameters	
Parameters for Isometric Plot	
Antenna Pattern Print/Plot Parameters	
Atmospheric Attenuation Parameters	
Rain Attenuation Parameters	
Jammer Parameters	
Execute - This should be the last one in each case	
Open a new data file	
Save this new window	
Close the new data file	
Open an existing data file	
Save this existing window	
Close this existing data file	
QUIT	

Figure 2 Main Menu



⊙

Signal Parameters

▽

Item 1

⊙ >

RF Transmit Frequency (MHz)

⊙ >

Item 3

⊙ >

Item 4

⊙ >

Peak Power (Mw)

⊙ >

Instaneous Bandwidth (MHz)

⊙ >

Cumulative System Losses (dB)

⊙ >

System Noise Temp. (K)

⊙ >

Transmit Pulse Width (micro sec)

⊙ >

Var 6

⊙ >

Figure 3 Signal Parameters

⊙
**Radar Detection Parameters**
▽

*0-Conolly's law, 1-Marcum/Swerling, 2-Shotland*

⊙ >

*0 - Fixed, # - Swerling Case*

⊙ >

*# - Independent Groups*

⊙ >

*# - Dependent Pulses in a Group*

⊙ >

*Var 1*

⊙ >

*Coherent Integration Gain*

⊙ >

*Probability of False Alarm (-10\*log(PFA))*

⊙ >

*Target Cross Section (sq. meters)*

⊙ >

*Detection Sensitivity (dB)*

⊙ >

*Detection Slope (default=1)*

⊙ >

Figure 4 Radar Detection Parameters

AEW Geometry	
Item 1	<input type="text"/>
Item 2	<input type="text"/>
Item 3	<input type="text"/>
Item 4	<input type="text"/>
Platform Altitude (ft.)	<input type="text"/>
Platform Pitch Angle (deg.)	<input type="text"/>
Platform Roll Angle (deg.)	<input type="text"/>
Var 4	<input type="text"/>
Var 5	<input type="text"/>
Var 6	<input type="text"/>

Figure 5 AEW Geometry



Target Range Limits	
Item 1	
Item 2	
Item 3	
Item 4	
Minimum Target Range (nmi)	
Maximum Target Range (nmi)	
Increment Range (nmi)	
Var 4	
Var 5	
Var 6	

Figure 7 Target Range Limits

Antenna Characteristics	
# - of Antenna Pattern Data Points	<input type="text"/>
Reference Count (10*full scale in dB)	<input type="text"/>
0 - Horizontal Polarity, 1 - Vertical Polarity	<input type="text"/>
0-T/R parameter, 1-Transmit pattern, 2-Receive pattern, 3-VPAT routing	<input type="text"/>
Antenna az. pointing angle (deg.)	<input type="text"/>
Antenna Reference Gain (dB)	<input type="text"/>
Min Angle of Antenna Pattern(if $11 \neq 0$ ), Vertical Beamwidth (deg)	<input type="text"/>
Increment of Pattern (if $11 \neq 0$ )	<input type="text"/>
Full Scale of Antenna Chart (dB)(if $11 \neq 0$ ), Elevation offset Angle (deg)	<input type="text"/>
Vertical Pattern Phase Parameter	<input type="text"/>

Figure 8 Antenna Characteristics



Stored Antenna Data Points	
# - of Current Antenna Point	<input type="text"/>
Recorder Count (10*value in dB)	<input type="text"/>
Item 3	<input type="text"/>
Item 4	<input type="text"/>
Voltage Pattern Sign (0, +1, -1)	<input type="text"/>
Var 2	<input type="text"/>
Var 3	<input type="text"/>
Var 4	<input type="text"/>
Var 5	<input type="text"/>
Var 6	<input type="text"/>

Figure 9 Antenna Data Points



Clutter Characteristics	
0-Land,1-Sea,Model Clutter Coeff func.2/3(L/S)-range, 4/5(L/S)-grazing	<input type="text" value="0"/> >1
Diffraction Option: 0-Interpolate,1-Multi Path,2-Fock	<input type="text" value="0"/> >0
0-(all cases except 2/3),Index# of Range Interval for coeff	<input type="text" value="0"/> >0
0-No Forward Scatter,1-Forward Reflection,Forward Power:2-Min,3-Ma	<input type="text" value="0"/> >1
Clutter Reflection Coeff(dB),(for cases 4/5 coeff taken at 10 (deg)	<input type="text" value="0"/> >-43.799999
Horizontal Beamwider(deg.)	<input type="text" value="0"/> >0.800000
Pulse Width (post-compressed) (micro sec.)	<input type="text" value="0"/> >0.200000
Wave Ht(ft):0-0',1-<1',2-(1-3)',3-(3-5)',4-(5-8)',5-(8-12)',6-(12-20)'	<input type="text" value="0"/> >5.000000
Min range (cases 2/3)(nmi), Slope of Coeff(case 4/5) (dB/decade)	<input type="text" value="0"/> >0.000000
Increment Range (cases 2/3) (nmi)	<input type="text" value="0"/> >0.000000

Figure 10 Clutter Characteristics




Cancellation Characteristics	
0-Single Cancellation ratio, #-Cancellation ratio for #th range value	<input type="text"/>
# - Pulses Per Antenna Beamwidth	<input type="text"/>
0-No Loss func., 1-Single Delay Canceller, 2-Double Delay Canceller	<input type="text"/>
Item 4	<input type="text"/>
Cancellation Ratio (dB)	<input type="text"/>
Threshold Loss Coefficient - A1 (if unknown use 0)	<input type="text"/>
Threshold Loss Coefficient - A2 (if unknown use 5)	<input type="text"/>
Threshold Loss Coefficient - B1 (if unknown use 1)	<input type="text"/>
Threshold Loss Coefficient - B2	<input type="text"/>
Threshold Loss Coefficient - B3	<input type="text"/>


Figure 11 Cancellation Characteristics

 *Title for Printout and Plot* 


*# of Title Line to follow (maximum is 4)*

 >


*Line one*

 >

*Line two*

 >

*Line three*

 >

*Line four*


 >

Figure 12 Title for Printout & Plot

Target Velocity Limits	
0-For Regular PD Calc, 1-For Cumulative PD Calc	
<input type="radio"/>	>
Item 2	
<input type="radio"/>	>
Item 3	
<input type="radio"/>	>
Item 4	
<input type="radio"/>	>
Minimum Target Range Rate (ft/sec)	
<input type="radio"/>	>
Maximum Target Range Rate (ft/sec)	
<input type="radio"/>	>
Increment Rate (ft/sec)	
<input type="radio"/>	>
Scan Interval (sec)	
<input type="radio"/>	>
Range Increment Between Printouts (nmi)	
<input type="radio"/>	>
Cumulative Prob Level for Special Print	
<input type="radio"/>	>

Figure 13 Target Velocity Limits

PD Contour Plot Data	
<i>0-Regular Plots only, 1-Automatic Contour Plots</i>	
<input type="radio"/>	>
<i># of Contour Levels (1+1/step_size)</i>	
<input type="radio"/>	>
<i>Item 3</i>	
<input type="radio"/>	>
<i>Item 4</i>	
<input type="radio"/>	>
<i>Maximum Target Altitude (ft.)</i>	
<input type="radio"/>	>
<i>Range Scale (nm/in.)</i>	
<input type="radio"/>	>
<i>Height Scale (ft/in.)</i>	
<input type="radio"/>	>
<i>Var 4</i>	
<input type="radio"/>	>
<i>Var 5</i>	
<input type="radio"/>	>
<i>Var 6</i>	
<input type="radio"/>	>

Figure 14 PD Contour Plot Data

Extended Surface Target Parameters	
0 - Header, 1 - Detail	
0 >	
0-Point Model, 1-Area Model, # of point (if 11=1)	
0 >	
Total # of detail points (detail cards must follow)	
0 >	
Item 4	
0 >	
Height (ft.)	
0 >	
Area (sq. ft.)	
0 >	
Var 3	
0 >	
Var 4	
0 >	
Var 5	
0 >	
Var 6	
0 >	

Figure 15 Extended Surface Target Parameters

Parameters for Isometric Plot		
Item 1: 0 - None (1,2,3)		1 - Signal/Noise
2 - Signal/(Clutter + Noise)		3 - Probability of Detection
Height Independent Plots: 10 - Reset to zero(11,12,13)		
11 - sh 1	12 - sh 2	13 - sh 3
Height Dependent Plots: 20 - Reset to zero(21,22,23)		
21 - sh1	22 - sh2	23 - sh3
Item 2: 0 - Reset		Height Independent Plots: 1 - Clutter/Noise
2 - Threshold Loss		3 - Cancellation Ratio
Height Dependent Plots: 1 - Clutter/Noise		2 - Signal/Noise
4 - Signal/(Clutter+Noise)		8 - Prob. of Detection
Item 3		
⊙ >		
Item 4		
⊙ >		
Range Scale (nm/in.)		
⊙ >		
dB or Pd scale (dB/in. or unit/in.)		
⊙ >		
Minimum dB or Pd (dB or unit)		
⊙ >		
Maximum Z Dimension (in.)		
⊙ >		
Height Scale (ft/in.)		
⊙ >		
Fixed XMAX for Plotting (nm)		
⊙ >		

Figure 16 Parameters for Isometric Plot





Atmospheric Attenuation Parameters	
0 - No Attenuation, 1 - Linear Attenuation, 2 - Asymtotic attenuation	0 >
0 - Reset, # of Element	0 >
variable NPOLY	0 >
Item 4	0 >
Attenuation value: (linear - dB/nmi) or (Asym. - dB/angle)	0 >
variable A	0 >
variable B	0 >
variable R	0 >
variable K	0 >
variable 4	0 >

Figure 18 Atmospheric Attenuation Parameters



Rain Attenuation Parameters	
	0 - None, 1 - Rain
<input type="radio"/>	>
	Item 2
<input type="radio"/>	>
	Item 3
<input type="radio"/>	>
	Item 4
<input type="radio"/>	>
	Attenuation (dB/nmi)
<input type="radio"/>	>
	Minimum Range (nmi)
<input type="radio"/>	>
	Maximum Range (nmi)
<input type="radio"/>	>
	var 4
<input type="radio"/>	>
	var 5
<input type="radio"/>	>
	var 6
<input type="radio"/>	>

Figure 19 Rain Attenuation Parameters

Jammer Parameters	
Item 1	<input type="text"/>
Item 2	<input type="text"/>
Item 3	<input type="text"/>
Item 4	<input type="text"/>
Jammer Effective Radiated Power (W)	<input type="text"/>
Jammer Bandwidth (MHz)	<input type="text"/>
Jammer Range (nmi)	<input type="text"/>
Antenna Sidelobe Gain (dB)	<input type="text"/>
SLC Cancellation Ratio (dB)	<input type="text"/>
Var 6	<input type="text"/>

Figure 20 Jammer Parameters

**Execute**

*0 - Clutter + Noise, 1 - Noise only*

☐ >

*0 - No pattern, print, plot, 1 - print, plot, pattern, continue!*

☐ >

*0 - Print, 1 - No Print*

☐ >

*Item 4*

☐ >

*Single Target Altitude (ft.)*

☐ >

*Var 2*

☐ >

*Var 3*

☐ >

*Var 4*

☐ >

*Var 5*

☐ >

*Var 6*

☐ >

Figure 21 Execution

Appendix B

SOURCE CODE LISTINGS

Tue Nov 29 14:02:23 1994

```
#ifndef INDATA_H
#define INDATA_H
#include <NRLx.h>
#include <NRLxutil.h>
#endif
```

Wed Sep 21 16:37:04 1994

```

/* This program uses NRLx Toolkit to create a window interface */
/* version of RADARPD input data file */
/* M. Yen Aug., 1993 */
/* Revision: M. Yen Sept., 1994 */
/* Recompiled and relinked the program with the upgraded NRLx library */
/* routines. */

#include "indata.h"

struct nrlx
{
    *console;
}

struct nrlx
{
    *popup2, *popup3, *popup4, *popup5, *popup6, *popup7,
    *popup8, *popup9, *popup10, *popup12, *popup13, *popup15,
    *popup16, *popup17, *popup18, *popup19, *popup20, *popup1,
    *popup11;

    *menu1,
    *entry_1, *entry_2, *entry_3, *entry_4, *entry_5, *entry_6,
    *entry_7, *entry_8, *entry_9, *entry_10, *entry_11, *entry_12,
    *entry_13, *entry_14, *entry_15, *entry_16, *entry_17,
    *entry_18, *entry_19;

    *entry_30,
    *entry_31,
    *entry_32,
    *entry_33;

    *entry_40, *entry_41, *entry_42;

    *text, *text1, *text2, *text3, *text4, *text5,
    *text200, *text201, *text202, *text203;

    *text6, *text7, *text8, *text9,
    *text10, *text11, *text12, *text13, *text14, *text204;

    *text15, *text16, *text17,
    *text205, *text206, *text207, *text208, *text209,
    *text210, *text211;

    *text18, *text19, *text20,
    *text212, *text213, *text214, *text215,
    *text216, *text217, *text218;

    *text21, *text22, *text23,
    *text219, *text220, *text221, *text222,
    *text223, *text224, *text225;

    *text24, *text25, *text26, *text27, *text31, *text32, *text33;
    *text28, *text29, *text30, *text35, *text36,
    *text226, *text227, *text228, *text229, *text230,
    *text231, *text232;

    *text37, *text38, *text39, *text40, *text41, *text42,
    *text43, *text44, *text45, *text46;

    *text47, *text48, *text49, *text50, *text51, *text52,
    *text53, *text54, *text55, *text233;
}

struct nrlx
{
    *text56, *text57, *text58, *text59, *text60, *text61,
    *text62, *text234, *text235, *text236;

    *text63, *text64, *text65, *text66, *text67,
    *text237, *text238, *text239, *text240, *text241;

    *text68, *text69, *text70, *text71, *text72,
    *text242, *text243, *text244, *text245, *text246;

    *text73, *text74, *text75, *text76, *text77, *text78, *text79,
    *text80, *text81, *text82, *text83, *text84, *text85, *text86,
    *text87, *text88, *text89, *text90, *text91, *text92, *text93,
    *text94, *text95, *text96, *text97, *text98,
    *text247, *text248;

    *text100, *text101, *text102, *text103, *text104, *text105,
    *text106, *text107, *text249, *text250;

    *text110, *text111, *text112, *text113, *text114, *text115,
    *text116, *text117, *text118, *text251;

    *text120, *text121, *text122, *text123, *text252, *text253,
    *text254, *text255, *text256, *text257;

    *text130, *text131, *text132, *text133, *text134,
    *text258, *text259, *text260, *text261, *text262;

    *text140, *text141, *text142, *text143, *text263,
    *text264, *text265, *text266, *text267, *text268;

    *text150, *text151, *text152, *text153, *text154;

    int ycount, inc, xcount, ycount;
    int cardno, loc, loc1, loc2, loc3, cndt;

    int i1, i2, i3, i4;
    float v1, v2, v3, v4, v5, v6;
    int line1, line2, line3, line4;

    char *str1, *str2, *str3, *str4;
    char *s11, *s12, *s13, *s14, *sv1, *sv2, *sv3, *sv4, *sv5, *sv6;
    char rstr1[79], rstr2[79], rstr3[79], rstr4[79];
    char astr1[79], astr2[79], astr3[79], astr4[79];

    void stoi1(), stoi2(), stoi3(), stoi4();
    void stof1(), stof2(), stof3(), stof4(), stof5(), stof6();

    void item16(), item2(), string1(), string2(), string3(), string4();

    FILE *fp, *fpl;
    char fname[30], infname[30];

    void main (argc, argv)
        int argc;
        char **argv;
    {
        void quit ();
    }
}

```

Wed Sep 21 16:37:04 1994

```

void signal();
void radarct();
void aewgeo();
void tarhgt();
void tarrng();
void antcha();
void antdata();
void cltchar();
void canchar();
void title();
void tarvelo();
void contour();
void surtar();
void isompt();
void antpat();
void atmattn();
void rainatt();
void jammer();
void execute();
void savewin();
void openfile();
void closefile();
void saveinf();
void openinfile();
void closeinfile();

str1=calloc(80,sizeof(char));
str2=calloc(80,sizeof(char));
str3=calloc(80,sizeof(char));
str4=calloc(80,sizeof(char));

s11=calloc(80,sizeof(char));
s12=calloc(80,sizeof(char));
s13=calloc(80,sizeof(char));
s14=calloc(80,sizeof(char));
s15=calloc(80,sizeof(char));
s16=calloc(80,sizeof(char));
s17=calloc(80,sizeof(char));
s18=calloc(80,sizeof(char));
s19=calloc(80,sizeof(char));
s20=calloc(80,sizeof(char));

console = NRLX (MainWindow, NULL,
XPos,10, YPos,10,
Label, "Input Data for RADARPD ",
Width, 500,
Height, 800,
BackgroundColor, "white",
FrameColor, "turquoise",
NULL);

menu1 = NRLX (Menu, console, XPos,10, YPos,10,
Label, "RADARPD Input Menu
Background, "black",
Foreground, "white",
Pinned, TRUE,
Textfont, "10x20",
NULL);

entry_1 = NRLX (Entry, menu1,
Label, "Signal Parameters",
Proc, signal,
Foreground, "black",
Background, "white",
Shape, Oval,
NULL);

entry_2 = NRLX (Entry, menu1,
Label, "Radar Detection Parameters",
Proc, radarct,
Foreground, "black",
Background, "white",
Shape, Oval,
NULL);

entry_3 = NRLX (Entry, menu1,
Label, "AEW Geometry",
Proc, aewgeo,
Foreground, "black",
Background, "white",
Shape, Oval,
NULL);

entry_4 = NRLX (Entry, menu1,
Label, "Target Height Limits",
Proc, tarhgt,
Foreground, "black",
Background, "white",
Shape, Oval,
NULL);

entry_5 = NRLX (Entry, menu1,
Label, "Target Range Limits",
Proc, tarrng,
Foreground, "black",
Background, "white",
Shape, Oval,
NULL);

entry_6 = NRLX (Entry, menu1,
Label, "Antenna Characteristics",
Proc, antcha,
Foreground, "black",
Background, "white",
Shape, Oval,
NULL);

entry_7 = NRLX (Entry, menu1,
Label, "Antenna Data Points",
Proc, antdata,
Foreground, "black",
Background, "white",
Shape, Oval,
NULL);

entry_8 = NRLX (Entry, menu1,
Label, "Clutter Characteristics",
Proc, cltchar,
Foreground, "black",
Background, "white",
Shape, Oval,
NULL);

```

```

entry_9 = NRLX (Entry, menu1,
Label, "Cancellation Characteristics",
Proc, canchar,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_19 = NRLX (Entry, menu1,
Label, "Title for printout & plot",
Proc, title, "black",
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_10 = NRLX (Entry, menu1,
Label, "Target Velocity Limits",
Proc, tarvelo,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_11 = NRLX (Entry, menu1,
Label, "PD Contour Plot Data",
Proc, contour,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_12 = NRLX (Entry, menu1,
Label, "Extended Surface Target Parameters",
Proc, surtar,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_13 = NRLX (Entry, menu1,
Label, "Parameters for Isometric Plot",
Proc, isompt,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_14 = NRLX (Entry, menu1,
Label, "Antenna Pattern Print/Plot Parameters",
Proc, antpat,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_15 = NRLX (Entry, menu1,
Label, "Atmospheric Attenuation Parameters",
Proc, atmattn,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_16 = NRLX (Entry, menu1,
Label, "Rain Attenuation Parameters",
Proc, rainatt,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_17 = NRLX (Entry, menu1,
Label, "Jammer Parameters",
Proc, jammer, "black",
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_18 = NRLX (Entry, menu1,
Label, "Execute - This should be the last one in each case",
Proc, execute,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_30 = NRLX (Entry, menu1,
Label, "Open a new data file",
Proc, openfile,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_31 = NRLX (Entry, menu1,
Label, "Save this new window",
Proc, savewin,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_32 = NRLX (Entry, menu1,
Label, "Close the new data file",
Proc, closefile,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

entry_40 = NRLX (Entry, menu1,
Label, "Open an existing data file",
Proc, openinfile,
Foreground, "black",
Background, "white",
Shape, oval,
NULL);

```



Wed Sep 21 16:37:04 1994

```

entry_42 = NRLx (Entry, menu1,
Label, "Save this in window",
Proc, saveinf,
Foreground, "black",
Background, "white",
Shape, Oval,
NULL);

entry_41 = NRLx (Entry, menu1,
Label, "Close this existing data file",
Proc, closeinfile,
Foreground, "black",
Background, "white",
Shape, Oval,
NULL);

entry_33 = NRLx (Entry, menu1,
Label, "QUIT",
Proc, quit,
Foreground, "black",
Background, "white",
Shape, Oval,
NULL);

NRLxStartInterface (console);

I
/* Popup Window 2 - Signal Parameters */
void signal()
{
    ycount=5, inc=70, xcount=10;
    cardno=2;
    i1=i2=i3=i4=0;
    v1=v2=v3=v4=v5=v6=0;

    popup2 = NRLx (Popup, console, XPos.530, YPos.20,
Label, "Signal Parameters",
Width, 600,
Height, 800,
Background, "white",
FrameColor, "red",
NULL);

    text200 = NRLx (TextEntry, popup2, XPos, xcount, YPos, ycount,
Label, "Item 1",
Width, 550,
Height, 25,
Proc, stoil,
NULL);

    text = NRLx (TextEntry, popup2, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "RF Transmit Frequency (MHz)",
Width, 550,
Height, 25,
Proc, stof5,
NULL);

    text201 = NRLx (TextEntry, popup2, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Item 3",
Width, 550,
Height, 25,
Proc, stoil3,
NULL);

    text202 = NRLx (TextEntry, popup2, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Item 4",
Width, 550,
Height, 25,
Proc, stof4,
NULL);

    text1 = NRLx (TextEntry, popup2, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Peak Power (Mw)",
Width, 550,
Height, 25,
Proc, stof1,
NULL);

    text2 = NRLx (TextEntry, popup2, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Instantaneous Bandwidth (MHz)",
Width, 550,
Height, 25,
Proc, stof2,
NULL);

    text3 = NRLx (TextEntry, popup2, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Cumulative system losses (dB)",
Width, 550,
Height, 25,
Proc, stof3,
NULL);

    text4 = NRLx (TextEntry, popup2, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "System Noise Temp. (K)",
Width, 550,
Height, 25,
Proc, stof4,
NULL);

    text5 = NRLx (TextEntry, popup2, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Transmit Pulse Width (micro sec)",
Width, 550,
Height, 25,
Proc, stof5,
NULL);

    text203 = NRLx (TextEntry, popup2, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "var 6",
Width, 550,
Height, 25,
Proc, stof6,
NULL);

```

Wed Sep 21 16:37:04 1994

```

/* read in an existing file, make changes on card 2 */
if (cndt == 1) {
    rewind (fpl);

    fscanf (fpl, "%d%d%d%d%d%f%f%f%f", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);

    while (cardno != 2) {
        fseek (fpl, 1, 1);
        fscanf (fpl, "%d%d%d%d%d%f%f%f%f", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
    }

    loc = ftell (fpl);
    printf("win 2 1st loc = %d\n", loc);
    loc = loc - 80;
    printf("win 2 2nd loc=%d\n", loc);

    printf (s11, "%d", i1);
    printf (s12, "%d", i2);
    printf (s13, "%d", i3);
    printf (s14, "%d", i4);
    printf (sv1, "%f", v1);
    printf (sv2, "%f", v2);
    printf (sv3, "%f", v3);
    printf (sv4, "%f", v4);
    printf (sv5, "%f", v5);
    printf (sv6, "%f", v6);

    NRLxLoadTextEntry (text200, s11);
    NRLxLoadTextEntry (text, s12);
    NRLxLoadTextEntry (text201, s13);
    NRLxLoadTextEntry (text202, s14);
    NRLxLoadTextEntry (text1, sv1);
    NRLxLoadTextEntry (text2, sv2);
    NRLxLoadTextEntry (text3, sv3);
    NRLxLoadTextEntry (text4, sv4);
    NRLxLoadTextEntry (text5, sv5);
    NRLxLoadTextEntry (text203, sv6);
}

/* Popup Window 3 - Radar Detection Parameters */
void radardet()
{
    ycount=5, inc=70, xcount=10;
    cardno=3;
    i1=i2=i3=i4=0;
    v1=v2=v3=v4=v5=v6=0;

    popup3 = NRLx (Popup, console, XPos, 530, YPos, 10,
        Label, "Radar Detection Parameters",
        Width, 600,
        Height, 800,

    text6 = NRLx (TextEntry, popup3, XPos, xcount, YPos, ycount,
        Background, "white",
        FrameColor, "purple",
        NULL);
    Label, "0-Conolly's law, 1-Marcum/Swerling, 2-Shotland ",
    Width, 530,
    Height, 25,
    Proc, stoil,
    NULL);

    text7 = NRLx (TextEntry, popup3, XPos, xcount, YPos, (ycount-ycount+inc),
        Label, "0 - Fixed, # - Swerling Case ",
        Width, 530,
        Height, 25,
        Proc, stoil2,
        NULL);

    text8 = NRLx (TextEntry, popup3, XPos, xcount, YPos, (ycount-ycount+inc),
        Label, "# - Independent Groups ",
        Width, 530,
        Height, 25,
        Proc, stoil3,
        NULL);

    text9 = NRLx (TextEntry, popup3, XPos, xcount, YPos, (ycount-ycount+inc),
        Label, "# - Dependent Pulses in a Group ",
        Width, 530,
        Height, 25,
        Proc, stoil4,
        NULL);

    text204 = NRLx (TextEntry, popup3, XPos, xcount, YPos, (ycount-ycount+inc),
        Label, "var 1",
        Width, 530,
        Height, 25,
        Proc, stof1,
        NULL);

    text10 = NRLx (TextEntry, popup3, XPos, xcount, YPos, (ycount-ycount+inc),
        Label, "Coherent Integration Gain ",
        Width, 530,
        Height, 25,
        Proc, stof2,
        NULL);

    text11 = NRLx (TextEntry, popup3, XPos, xcount, YPos, (ycount-ycount+inc),
        Label, "Probability of False Alarm (-10*log(PFA))",
        Width, 530,
        Height, 25,
        Proc, stof3,
        NULL);

    text12 = NRLx (TextEntry, popup3, XPos, xcount, YPos, (ycount-ycount+inc),
        Label, "Target Cross Section (sq. meters)",
        Width, 530,
        Height, 25,
        Proc, stof4,
        NULL);

    text13 = NRLx (TextEntry, popup3, XPos, xcount, YPos, (ycount-ycount+inc),

```

```

Label, "Detection Sensitivity (db)",
Width, 550,
Height, 25,
Proc, stof5,
NULL);

text14 = NRLX (TextEntry, popup3, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Detection Slope (default=1)",
Width, 550,
Height, 25,
Proc, stof6,
NULL);

/* read in an existing file, make changes on card 3 */
if (cndt == 1) {
    rewind (fp1);
    fscanf (fp1, "%d%d%d%d%d%d", &cardno, &si1, &si2, &si3, &si4, &v
1, &v2, &v3, &v4, &v5, &v6);
    while (cardno != 3) {
        fseek (fp1, 1, 1);
        fscanf (fp1, "%d%d%d%d%d%d", &cardno, &si1, &si2, &si3, &si4, &v
1, &v2, &v3, &v4, &v5, &v6);
    }

    loc = ftell (fp1);
    printf("win 3 1st loc = %d\n", loc);
    loc = loc - 80;
    printf("win 3 2nd loc=%d\n", loc);

    printf (si1, "%d", i1);
    printf (si2, "%d", i2);
    printf (si3, "%d", i3);
    printf (si4, "%d", i4);
    printf (sv1, "%f", v1);
    printf (sv2, "%f", v2);
    printf (sv3, "%f", v3);
    printf (sv4, "%f", v4);
    printf (sv5, "%f", v5);
    printf (sv6, "%f", v6);

    NRLLoadTextEntry (text6, si1);
    NRLLoadTextEntry (text7, si2);
    NRLLoadTextEntry (text8, si3);
    NRLLoadTextEntry (text9, si4);
    NRLLoadTextEntry (text10, sv1);
    NRLLoadTextEntry (text11, sv2);
    NRLLoadTextEntry (text12, sv3);
    NRLLoadTextEntry (text13, sv4);
    NRLLoadTextEntry (text14, sv5);
}

}

/* Popup Window 4 - AEW Geometry */
void aewgeo()

```

```

Proc, stof3,
NULL);

text209 = NRLX (TextEntry, popup4, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Var 4",
Width, 550,
Height, 25,
Proc, stof4,
NULL);

text210 = NRLX (TextEntry, popup4, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Var 5",
Width, 550,
Height, 25,
Proc, stof5,
NULL);

text211 = NRLX (TextEntry, popup4, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Var 6",
Width, 550,
Height, 25,
Proc, stof6,
NULL);

/* read in an existing file, make changes on card 4*/
if (cndt == 1) {
rewind (fpl);

fscanf (fpl, "%d%d%d%d%d%f%f%f", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);

while (cardno !=4) {
fseek (fpl, 1, 1);
fscanf (fpl, "%d%d%d%d%d%f%f%f", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
}

loc = ftell (fpl);
printf("win 4 1st loc = %d\n", loc);
loc = loc - 80;
printf("win 4 2nd loc=%d\n", loc);

printf (s11, "%d", i1);
printf (s12, "%d", i2);
printf (s13, "%d", i3);
printf (s14, "%d", i4);
printf (sv1, "%f", v1);
printf (sv2, "%f", v2);
printf (sv3, "%f", v3);
printf (sv4, "%f", v4);
printf (sv5, "%f", v5);
printf (sv6, "%f", v6);

NRLXLoadTextEntry (text205, s11);
NRLXLoadTextEntry (text206, s12);
NRLXLoadTextEntry (text207, s13);
NRLXLoadTextEntry (text208, s14);
NRLXLoadTextEntry (text15, sv1);
NRLXLoadTextEntry (text16, sv2);
NRLXLoadTextEntry (text17, sv3);
NRLXLoadTextEntry (text209, sv4);
NRLXLoadTextEntry (text210, sv5);
NRLXLoadTextEntry (text211, sv6);
}

/* Popup Window 5 - Target Height Limits */
void tarhgt()
{
ycount=5, inc=70, xcount=10;
cardno=5;
i1=i2=i3=i4=0;
v1=v2=v3=v4=v5=v6=0;

popup5 = NRLX (Popup, console, XPos, 530, YPos, 20,
Label, "Target Height Limits",
Width, 600,
Height, 800,
Background, "white",
FrameColor, "gold",
NULL);

text212 = NRLX (TextEntry, popup5, XPos, xcount, YPos, ycount,
Label, "Item 1",
Width, 550,
Height, 25,
Proc, stof1,
NULL);

text213 = NRLX (TextEntry, popup5, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Item 2",
Width, 550,
Height, 25,
Proc, stof2,
NULL);

text214 = NRLX (TextEntry, popup5, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Item 3",
Width, 550,
Height, 25,
Proc, stof3,
NULL);

text215 = NRLX (TextEntry, popup5, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Item 4",
Width, 550,
Height, 25,
Proc, stof4,
NULL);

text18 = NRLX (TextEntry, popup5, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Minimum Target Altitude (ft.)",
Width, 550,
Height, 25,
Proc, stof1,
NULL);

```



Wed Sep 21 16:37:04 1994

```

        width, 550,
        height, 25,
        proc, stof14,
        NULL);

text21 = NRLX (TextEntry, popup6, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Minimum Target Range (nmi)",
width, 550,
height, 25,
proc, stof1,
NULL);

text22 = NRLX (TextEntry, popup6, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Maximum Target Range (nmi)",
width, 550,
height, 25,
proc, stof2,
NULL);

text23 = NRLX (TextEntry, popup6, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Increment Range (nmi)",
width, 550,
height, 25,
proc, stof3,
NULL);

text223 = NRLX (TextEntry, popup6, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Var 4",
width, 550,
height, 25,
proc, stof4,
NULL);

text224 = NRLX (TextEntry, popup6, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Var 5",
width, 550,
height, 25,
proc, stof5,
NULL);

text225 = NRLX (TextEntry, popup6, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Var 6",
width, 550,
height, 25,
proc, stof6,
NULL);

/* read in an existing file, make changes on card 6 */
if (cndt == 1) {
    rewind (fpl);

    fscanf (fpl, "%d%d%d%d%d%f%f%f%f", &cardno, &sil, &si2, &si3, &si4, &v
1, &v2, &v3, &v4, &v5, &v6);

    while (cardno != 6) {
        fseek (fpl, 1, 1);
        fscanf (fpl, "%d%d%d%d%d%f%f%f%f", &cardno, &sil, &si2, &si3, &si4, &v
1, &v2, &v3, &v4, &v5, &v6);
    }

    width, 550,
    height, 25,
    proc, stof14,
    NULL);

    printf("win 6 1st loc = %d\n", loc);
    loc = loc - 80;
    printf("win 6 2nd loc=%d\n", loc);

    printf (sil, "%d", il);
    printf (si2, "%d", i2);
    printf (si3, "%d", i3);
    printf (si4, "%d", i4);
    printf (sv1, "%f", v1);
    printf (sv2, "%f", v2);
    printf (sv3, "%f", v3);
    printf (sv4, "%f", v4);
    printf (sv5, "%f", v5);
    printf (sv6, "%f", v6);

    NRLXLoadTextEntry (text219, sil);
    NRLXLoadTextEntry (text220, si2);
    NRLXLoadTextEntry (text221, si3);
    NRLXLoadTextEntry (text222, si4);
    NRLXLoadTextEntry (text221, sv1);
    NRLXLoadTextEntry (text22, sv2);
    NRLXLoadTextEntry (text23, sv3);
    NRLXLoadTextEntry (text223, sv4);
    NRLXLoadTextEntry (text224, sv5);
    NRLXLoadTextEntry (text225, sv6);
}

/* Popup Window 7 - Antenna Characteristics */
void antcha()
{
    ycount=5, inc=70, xcount=10;
    cardno=7;
    il=i2=i3=i4=0;
    v1=v2=v3=v4=v5=v6=0;

    popup7 = NRLX (Popup, console, xpos, 530, ypos, 10,
Label, "Antenna Characteristics",
width, 600,
height, 800,
background, "white",
framecolor, "dark turquoise",
NULL);

    text24 = NRLX (TextEntry, popup7, xpos, xcount, ypos, ycount,
Label, "# of Antenna Pattern Data Points ",
width, 550,
height, 25,
proc, stof1,
NULL);

    text25 = NRLX (TextEntry, popup7, xpos, xcount, ypos, (ycount=ycount+inc),
Label, "Reference Count (10*full scale in dB)",
width, 550,
height, 25,
proc, stof1,
NULL);

```

Wed Sep 21 16:37:04 1994

```

Proc, stoi2,
NULL);

text26 = NRLX (TextEntry, popup7, XPos, xcount, YPos, (Ycount=Ycount+inc),
Label, "0 - Horizontal Polarity, 1 - Vertical Polarity",
Width, 550,
Height, 25,
Proc, stoi3,
NULL);

text27 = NRLX (TextEntry, popup7, XPos, xcount, YPos, (Ycount=Ycount+inc),
Label, "0-1/R parameter, 1-Transmit pattern, 2-Receive pattern, 3-VPA
T routine",
Width, 550,
Height, 25,
Proc, stoi4,
NULL);

text28 = NRLX (TextEntry, popup7, XPos, xcount, YPos, (Ycount=Ycount+inc),
Label, "Antenna az. pointing angle (deg.)",
Width, 550,
Height, 25,
Proc, stof1,
NULL);

text29 = NRLX (TextEntry, popup7, XPos, xcount, YPos, (Ycount=Ycount+inc),
Label, "Antenna Reference Gain (dB)",
Width, 550,
Height, 25,
Proc, stof2,
NULL);

text30 = NRLX (TextEntry, popup7, XPos, xcount, YPos, (Ycount=Ycount+inc),
Label, "Min Angle of Antenna Pattern (if il=0), Vertical Beamwid
h (deg)",
Width, 550,
Height, 25,
Proc, stof3,
NULL);

text31 = NRLX (TextEntry, popup7, XPos, xcount, YPos, (Ycount=Ycount+inc),
Label, "Increment of Pattern (if il=0)",
Width, 550,
Height, 25,
Proc, stof4,
NULL);

text32 = NRLX (TextEntry, popup7, XPos, xcount, YPos, (Ycount=Ycount+inc),
Label, "Full Scale of Antenna Chart (dB) (if il=0), Elevation off
set Angle (deg)",
Width, 550,
Height, 25,
Proc, stof5,
NULL);

text33 = NRLX (TextEntry, popup7, XPos, xcount, YPos, (Ycount=Ycount+inc),
Label, "Vertical Pattern Phase Parameter",
Width, 550,
Height, 25,
Proc, stof6,
NULL);

/* read in an existing file, make changes on card 7*/
if (cndt == 1) {
rewind (fpl);
fscanf (fpl, "%d%d%d%d%d%f%f%f%f", &cardno, &il, &si2, &si3, &si4, &v
1, &sv2, &sv3, &sv4, &sv5, &sv6);

while (cardno != 7) {
fseek (fpl, 1, 1);
fscanf (fpl, "%d%d%d%d%d%f%f%f%f", &cardno, &il, &si2, &si3, &si4, &v
1, &sv2, &sv3, &sv4, &sv5, &sv6);
}

loc = ftell (fpl);
printf ("win 7 1st loc = %d\n", loc);
loc = loc - 80;
printf ("win 7 2nd loc=%d\n", loc);

printf (sil, "%d", il);
printf (si2, "%d", i2);
printf (si3, "%d", i3);
printf (si4, "%d", i4);
printf (sv1, "%f", v1);
printf (sv2, "%f", v2);
printf (sv3, "%f", v3);
printf (sv4, "%f", v4);
printf (sv5, "%f", v5);
printf (sv6, "%f", v6);

NRLXLoadTextEntry (text24, sil);
NRLXLoadTextEntry (text25, si2);
NRLXLoadTextEntry (text26, si3);
NRLXLoadTextEntry (text27, si4);
NRLXLoadTextEntry (text28, sv1);
NRLXLoadTextEntry (text29, sv2);
NRLXLoadTextEntry (text30, sv3);
NRLXLoadTextEntry (text31, sv4);
NRLXLoadTextEntry (text32, sv5);
NRLXLoadTextEntry (text33, sv6);
}

/* Popup Window 8 - Stored Antenna Data Points */
void antdata()
{
Ycount=5, inc=70, xcount=10;
cardno=8;
il=i2=i3=i4=0;
v1=v2=v3=v4=v5=v6=0;

popup8 = NRLX (Popup, console, XPos, 530, YPos, 20,
Label, "Stored Antenna Data Points",
Width, 600,
Height, 800,
Background, "White",
}

```

Wed Sep 21 16:37:04 1994

```

FrameColor, "violet red",
NULL);

text34 = NRLX (TextEntry, popup8, XPos, xcount, YPos, ycount,
Label, "# - of Current Antenna Point",
Width, 550,
Height, 25,
Proc, stof5,
NULL);

text35 = NRLX (TextEntry, popup8, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Recorder Count (10*value in dB)",
Width, 550,
Height, 25,
Proc, stof2,
NULL);

text226 = NRLX (TextEntry, popup8, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Item 3",
Width, 550,
Height, 25,
Proc, stof3,
NULL);

text227 = NRLX (TextEntry, popup8, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Item 4",
Width, 550,
Height, 25,
Proc, stof4,
NULL);

text36 = NRLX (TextEntry, popup8, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Voltage Pattern Sign (0, +1, -1)",
Width, 550,
Height, 25,
Proc, stof1,
NULL);

text228 = NRLX (TextEntry, popup8, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Var 2",
Width, 550,
Height, 25,
Proc, stof2,
NULL);

text229 = NRLX (TextEntry, popup8, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Var 3",
Width, 550,
Height, 25,
Proc, stof3,
NULL);

text230 = NRLX (TextEntry, popup8, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Var 4",
Width, 550,
Height, 25,
Proc, stof4,
NULL);

text231 = NRLX (TextEntry, popup8, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Var 5",

```

```

Width, 550,
Height, 25,
Proc, stof5,
NULL);

text232 = NRLX (TextEntry, popup8, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Var 6",
Width, 550,
Height, 25,
Proc, stof6,
NULL);

/* read in an existing file, make changes */
if (cndt == 1) {
    rewind (fp1);
    fscanf (fp1, "%d%d%d%d%d%d%f%f%f", scardno, si1, si2, si3, si4, sv
1, sv2, sv3, sv4, sv5, sv6);
    while (cardno != 8) {
        fseek (fp1, 1, 1);
        fscanf (fp1, "%d%d%d%d%d%d%f%f%f", scardno, si1, si2, si3, si4, sv
1, sv2, sv3, sv4, sv5, sv6);
    }
    loc = ftell (fp1);
    printf("win 8 1st loc = %d\n", loc);
    loc = loc - 80;
    printf("win 8 2nd loc=%d\n", loc);
    printf (si1, "%d", i1);
    printf (si2, "%d", i2);
    printf (si3, "%d", i3);
    printf (si4, "%d", i4);
    printf (sv1, "%f", v1);
    printf (sv2, "%f", v2);
    printf (sv3, "%f", v3);
    printf (sv4, "%f", v4);
    printf (sv5, "%f", v5);
    printf (sv6, "%f", v6);
    NRLXLoadTextEntry (text34, si1);
    NRLXLoadTextEntry (text35, si2);
    NRLXLoadTextEntry (text226, si3);
    NRLXLoadTextEntry (text227, si4);
    NRLXLoadTextEntry (text36, sv1);
    NRLXLoadTextEntry (text228, sv2);
    NRLXLoadTextEntry (text229, sv3);
    NRLXLoadTextEntry (text230, sv4);
    NRLXLoadTextEntry (text231, sv5);
    NRLXLoadTextEntry (text232, sv6);
}

}

/* Popup Window 9 - Clutter Characteristics */
void clutter()
{

```



Wed Sep 21 16:37:04 1994

```

ycount=5, inc=70, xcount=3;
cardno=9;
il=i2=i3=i4=0;
v1=v2=v3=v4=v5=v6=0;

popup9 = NRLx (Popup, console, XPos, 530, YPos, 20,
Label, "Clutter Characteristics",
Width, 600,
Height, 800,
Background, "White",
FrameColor, "blue",
NULL);

text37 = NRLx (TextEntry, popup9, XPos, xcount, YPos, ycount,
(L/S)-grazing angle",
Label, "0-Land, 1-Sea, Model Clutter Coeff func: 2/3(L/S)-range, 4/5
Width, 580,
Height, 25,
Proc, stof1,
NULL);

text38 = NRLx (TextEntry, popup9, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Diffraction Option: 0-Interpolate, 1-Multi Path, 2-Fock",
Width, 580,
Height, 25,
NULL);

text39 = NRLx (TextEntry, popup9, XPos, xcount, YPos, (ycount-ycount+inc),
ff",
Label, "0-(all cases except 2/3), index# of Range Interval for coe
Width, 580,
Height, 25,
Proc, stof3,
NULL);

text40 = NRLx (TextEntry, popup9, XPos, xcount, YPos, (ycount-ycount+inc),
-Min, 3-Max",
Label, "0-No Forward Scatter, 1-Forward Reflection, Forward Power: 2
Width, 580,
Height, 25,
Proc, stof4,
NULL);

text41 = NRLx (TextEntry, popup9, XPos, xcount, YPos, (ycount-ycount+inc),
at 10 (deg)",
Label, "Clutter Reflection Coeff (dB), (for cases 4/5 coeff taken
Width, 580,
Height, 25,
Proc, stof1,
NULL);

text42 = NRLx (TextEntry, popup9, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Horizontal Beamwidth (deg.)",
Width, 580,
Height, 25,
Proc, stof2,
NULL);

text43 = NRLx (TextEntry, popup9, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Pulse Width (post-compressed) (micro sec.)",
Width, 580,
Height, 25,
Proc, stof3,
NULL);

width, 580,
Height, 25,
Proc, stof3,
NULL);

text44 = NRLx (TextEntry, popup9, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Wave Ht(ft): 0-0', 1-1', 2-(1-3)', 3-(3-5)', 4-(5-8)', 5-(8-12)
Width, 580,
Height, 25,
Proc, stof4,
NULL);

text45 = NRLx (TextEntry, popup9, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Min range (cases 2/3)(nmi), Slope of Coeff (case 4/5) (dB
/decade)",
Width, 580,
Height, 25,
Proc, stof5,
NULL);

text46 = NRLx (TextEntry, popup9, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Increment Range (cases 2/3) (nmi)",
Width, 580,
Height, 25,
Proc, stof6,
NULL);

/* read in an existing file, make changes on card 9 */
if (cndt == 1) {
    rewind (fp1);
    fscanf (fp1, "%d%d%d%d%d%d%d%d%d", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
    while (cardno != 9) {
        fseek (fp1, 1, 1);
        fscanf (fp1, "%d%d%d%d%d%d%d%d%d", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
    }
    loc = ftell (fp1);
    printf("win 9 1st loc = %d\n", loc);
    loc = loc - 80;
    printf("win 9 2nd loc = %d\n", loc);
    printf (s11, "%d", i1);
    printf (s12, "%d", i2);
    printf (s13, "%d", i3);
    printf (s14, "%d", i4);
    printf (sv1, "%f", v1);
    printf (sv2, "%f", v2);
    printf (sv3, "%f", v3);
    printf (sv4, "%f", v4);
    printf (sv5, "%f", v5);
    printf (sv6, "%f", v6);

```

```

NRLxLoadTextEntry (text37, si1);
NRLxLoadTextEntry (text38, si2);
NRLxLoadTextEntry (text39, si3);
NRLxLoadTextEntry (text40, si4);
NRLxLoadTextEntry (text41, sv1);
NRLxLoadTextEntry (text42, sv2);
NRLxLoadTextEntry (text43, sv3);
NRLxLoadTextEntry (text44, sv4);
NRLxLoadTextEntry (text45, sv5);
NRLxLoadTextEntry (text46, sv6);
}

/* PopUp Window 10 - Cancellation Characteristics */
void canchar()
{
    ycount=5, inc=70, xcount=3;
    cardno=10;
    i1=i2=i3=i4=0;
    v1=v2=v3=v4=v5=v6=0;

    popup10 = NRLx (Popup, console, XPos, 530, YPos, 20,
        Label, "Cancellation Characteristics",
        Width, 600,
        Height, 800,
        Background, "white",
        FrameColor, "medium sea green",
        NULL);

    text47 = NRLx (TextEntry, popup10, XPos, xcount, YPos, ycount,
        range value",
        Label, "0-Single Cancellation ratio, #-cancellation ratio for #th",
        Width, 580,
        Height, 25,
        Proc, sto11,
        NULL);

    text48 = NRLx (TextEntry, popup10, XPos, xcount, YPos, (ycount=ycount+inc),
        Label, "# - Pulses per Antenna Beamwidth",
        Width, 580,
        Height, 25,
        Proc, sto12,
        NULL);

    text49 = NRLx (TextEntry, popup10, XPos, xcount, YPos, (ycount=ycount+inc),
        canceller",
        Label, "0-No Loss func., 1-Single Delay Cancellation, 2-Double Delay c",
        Width, 580,
        Height, 25,
        Proc, sto13,
        NULL);

    text233 = NRLx (TextEntry, popup10, XPos, xcount, YPos, (ycount=ycount+inc),
        Label, "Item 4",
        Width, 580,
        Height, 25,
        Proc, sto14,
        NULL);

    text50 = NRLx (TextEntry, popup10, XPos, xcount, YPos, (ycount=ycount+inc),
        Label, "Cancellation Ratio (dB)",
        Width, 580,
        Height, 25,
        Proc, stof1,
        NULL);

    text51 = NRLx (TextEntry, popup10, XPos, xcount, YPos, (ycount=ycount+inc),
        Label, "Threshold Loss Coefficient - A1 (if unknown use 0)",
        Width, 580,
        Height, 25,
        Proc, stof2,
        NULL);

    text52 = NRLx (TextEntry, popup10, XPos, xcount, YPos, (ycount=ycount+inc),
        Label, "Threshold Loss Coefficient - A2 (if unknown use 5)",
        Width, 580,
        Height, 25,
        Proc, stof3,
        NULL);

    text53 = NRLx (TextEntry, popup10, XPos, xcount, YPos, (ycount=ycount+inc),
        Label, "Threshold Loss Coefficient - B1 (if unknown use 1)",
        Width, 580,
        Height, 25,
        Proc, stof4,
        NULL);

    text54 = NRLx (TextEntry, popup10, XPos, xcount, YPos, (ycount=ycount+inc),
        Label, "Threshold Loss Coefficient - B2",
        Width, 580,
        Height, 25,
        Proc, stof5,
        NULL);

    text55 = NRLx (TextEntry, popup10, XPos, xcount, YPos, (ycount=ycount+inc),
        Label, "Threshold Loss Coefficient - B3",
        Width, 580,
        Height, 25,
        Proc, stof6,
        NULL);

    /* read in an existing file, make changes on card 10*/
    if (cndt == 1) {
        rewind (fp1);

        fscanf (fp1, "%d%d%d%d%d%d", &scardno, &si1, &si2, &si3, &si4, &sv
            1, &sv2, &sv3, &sv4, &sv5, &sv6);

        while (cardno != 10) {
            fseek (fp1, 1, 1);
            fscanf (fp1, "%d%d%d%d%d%d", &scardno, &si1, &si2, &si3, &si4, &sv
                1, &sv2, &sv3, &sv4, &sv5, &sv6);
        }
    }

```

Wed Sep 21 16:37:05 1994

```

loc = fcall (fpl);
printf("win 10 1st loc = %d\n", loc);
loc = loc - 80;
printf("win 10 2nd loc=%d\n", loc);

printf (s1, "%d", i1);
printf (s2, "%d", i2);
printf (s3, "%d", i3);
printf (s4, "%d", i4);
printf (sv1, "%f", v1);
printf (sv2, "%f", v2);
printf (sv3, "%f", v3);
printf (sv4, "%f", v4);
printf (sv5, "%f", v5);
printf (sv6, "%f", v6);

NRLXLoadTextEntry (text47, s1);
NRLXLoadTextEntry (text48, s2);
NRLXLoadTextEntry (text49, s3);
NRLXLoadTextEntry (text233, s4);
NRLXLoadTextEntry (text50, sv1);
NRLXLoadTextEntry (text51, sv2);
NRLXLoadTextEntry (text52, sv3);
NRLXLoadTextEntry (text53, sv4);
NRLXLoadTextEntry (text54, sv5);
NRLXLoadTextEntry (text55, sv6);
}

/* Popup Window 12 - Target Velocity Limits */
void tarvelo()
{
    ycount=5, inc=70, xcount=10;
    cardno=12;
    i1=i2=i3=i4=0;
    v1=v2=v3=v4=v5=v6=0;

    popup12 = NRLX (Popup, console, xpos,530, ypos, 20,
        Label, "Target Velocity Limits",
        Width, 600,
        Height, 800,
        Background, "White",
        FrameColor, "Dark Orchid",
        NULL);

    text56 = NRLX (TextEntry, popup12, xpos, xcount, ypos, ycount,
        Label, "0-For Regular PD Calc, 1-For Cumulative PD Calc",
        Width, 580,
        Height, 25,
        Proc, stoil,
        NULL);

    text234 = NRLX (TextEntry, popup12, xpos, xcount, ypos, (ycount-ycount+inc),
        Label, "Item 2",
        Width, 580,
        Height, 25,
        Proc, stoil2,
        NULL);

    loc = fcall (fpl);
    printf("win 10 1st loc = %d\n", loc);
    loc = loc - 80;
    printf("win 10 2nd loc=%d\n", loc);

    text235 = NRLX (TextEntry, popup12, xpos, xcount, ypos, (ycount-ycount+inc),
        Label, "Item 3",
        Width, 580,
        Height, 25,
        Proc, stoil3,
        NULL);

    text236 = NRLX (TextEntry, popup12, xpos, xcount, ypos, (ycount-ycount+inc),
        Label, "Item 4",
        Width, 580,
        Height, 25,
        Proc, stoil4,
        NULL);

    text57 = NRLX (TextEntry, popup12, xpos, xcount, ypos, (ycount-ycount+inc),
        Label, "Minimum Target Range Rate (ft/sec)",
        Width, 580,
        Height, 25,
        Proc, stof1,
        NULL);

    text58 = NRLX (TextEntry, popup12, xpos, xcount, ypos, (ycount-ycount+inc),
        Label, "Maximum Target Range Rate (ft/sec)",
        Width, 580,
        Height, 25,
        Proc, stof2,
        NULL);

    text59 = NRLX (TextEntry, popup12, xpos, xcount, ypos, (ycount-ycount+inc),
        Label, "Increment Rate (ft/sec)",
        Width, 580,
        Height, 25,
        Proc, stof3,
        NULL);

    text60 = NRLX (TextEntry, popup12, xpos, xcount, ypos, (ycount-ycount+inc),
        Label, "Scan Interval (sec)",
        Width, 580,
        Height, 25,
        Proc, stof4,
        NULL);

    text61 = NRLX (TextEntry, popup12, xpos, xcount, ypos, (ycount-ycount+inc),
        Label, "Range Increment Between Printouts (nml)",
        Width, 580,
        Height, 25,
        Proc, stof5,
        NULL);

    text62 = NRLX (TextEntry, popup12, xpos, xcount, ypos, (ycount-ycount+inc),
        Label, "Cumulative Prob Level for Special Print",
        Width, 580,
        Height, 25,
        Proc, stof6,
        NULL);

    /* read in an existing file, make changes on card 12 */
    if (cndt == 1) {

```

Wed Sep 21 16:37:05 1994

```

rewind (fpl);
    fscanf (fpl, "%d%d%d%d%d%f%f%f%f", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
    while (cardno != 12) {
        fseek (fpl, 1, 1);
        fscanf (fpl, "%d%d%d%d%d%f%f%f%f", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
    }
    loc = ftell (fpl);
    printf("win 12 1st loc = %d\n", loc);
    loc = loc - 80;
    printf("win 12 2nd loc=%d\n", loc);
    printf (s11, "%d", i1);
    printf (s12, "%d", i2);
    printf (s13, "%d", i3);
    printf (s14, "%d", i4);
    printf (sv1, "%f", v1);
    printf (sv2, "%f", v2);
    printf (sv3, "%f", v3);
    printf (sv4, "%f", v4);
    printf (sv5, "%f", v5);
    printf (sv6, "%f", v6);

    NRLXLoadTextEntry (text56, s11);
    NRLXLoadTextEntry (text234, s12);
    NRLXLoadTextEntry (text235, s13);
    NRLXLoadTextEntry (text236, s14);
    NRLXLoadTextEntry (text37, sv1);
    NRLXLoadTextEntry (text38, sv2);
    NRLXLoadTextEntry (text59, sv3);
    NRLXLoadTextEntry (text60, sv4);
    NRLXLoadTextEntry (text61, sv5);
    NRLXLoadTextEntry (text62, sv6);
}

/* Popup Window 13 - PD Contour Plot Data */
void contour()
{
    ycount=5, inc=70, xcount=10;
    cardno=13;
    i1=i2=i3=i4=0;
    v1=v2=v3=v4=v5=v6=0;

    popup13 = NRLX (Popup, console, XPos, 530, YPos, 20,
        Label, "PD Contour Plot Data",
        Width, 600,
        Height, 800, "White",
        Background, "Brown",
        FrameColor, "Brown",
        NULL);

    text63 = NRLX (TextEntry, popup13, XPos, xcount, YPos, ycount,
        Label, "0-Regular Plots only, 1-Automatic Contour Plots",
        Width, 580,
        Height, 25,
        Proc, stoil,
        NULL);

    text64 = NRLX (TextEntry, popup13, XPos, xcount, YPos, (ycount-ycount+inc),
        Label, "# of Contour Levels (1+1/step_size)",
        Width, 580,
        Height, 25,
        Proc, stof2,
        NULL);

    text237 = NRLX (TextEntry, popup13, XPos, xcount, YPos, (ycount-ycount+inc),
        Label, "item 3",
        Width, 580,
        Height, 25,
        Proc, stof3,
        NULL);

    text238 = NRLX (TextEntry, popup13, XPos, xcount, YPos, (ycount-ycount+inc),
        Label, "item 4",
        Width, 580,
        Height, 25,
        Proc, stof4,
        NULL);

    text65 = NRLX (TextEntry, popup13, XPos, xcount, YPos, (ycount-ycount+inc),
        Label, "Maximum Target Altitude (ft.)",
        Width, 580,
        Height, 25,
        Proc, stof1,
        NULL);

    text66 = NRLX (TextEntry, popup13, XPos, xcount, YPos, (ycount-ycount+inc),
        Label, "Range Scale (nmi/in.)",
        Width, 580,
        Height, 25,
        Proc, stof2,
        NULL);

    text67 = NRLX (TextEntry, popup13, XPos, xcount, YPos, (ycount-ycount+inc),
        Label, "Height Scale (ft/in.)",
        Width, 580,
        Height, 25,
        Proc, stof3,
        NULL);

    text239 = NRLX (TextEntry, popup13, XPos, xcount, YPos, (ycount-ycount+inc),
        Label, "Var 4",
        Width, 580,
        Height, 25,
        Proc, stof4,
        NULL);

    text240 = NRLX (TextEntry, popup13, XPos, xcount, YPos, (ycount-ycount+inc),
        Label, "Var 5",
        Width, 580,
        Height, 25,

```

```

Proc, stof5,
NULL);

text241 = NRLX (TextEntry, popup13, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Var 6",
Width, 580,
Height, 25,
Proc, stof6,
NULL);

/* read in an existing file, make changes on card 13*/
if (cndt == 1) {
rewind (fpl);

fscanf (fpl, "%d%d%d%d%d%f%f", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);

while (cardno != 13) {
fseek (fpl, 1, 1);
fscanf (fpl, "%d%d%d%d%d%f%f", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
}

loc = ftell (fpl);
printf("win 13 list loc = %d\n", loc);
loc = loc - 80;
printf("win 13 2nd loc=%d\n", loc);

printf (s1, "%d", i1);
printf (s2, "%d", i2);
printf (s3, "%d", i3);
printf (s4, "%d", i4);
printf (sv1, "%f", v1);
printf (sv2, "%f", v2);
printf (sv3, "%f", v3);
printf (sv4, "%f", v4);
printf (sv5, "%f", v5);
printf (sv6, "%f", v6);

NRLXLoadTextEntry (text63, s1);
NRLXLoadTextEntry (text64, s2);
NRLXLoadTextEntry (text237, s3);
NRLXLoadTextEntry (text238, s4);
NRLXLoadTextEntry (text65, sv1);
NRLXLoadTextEntry (text66, sv2);
NRLXLoadTextEntry (text67, sv3);
NRLXLoadTextEntry (text239, sv4);
NRLXLoadTextEntry (text240, sv5);
NRLXLoadTextEntry (text241, sv6);
}

/* Popup Window 15 - Extended Surface Target Parameters */
void surtar()
{
ycount=5, inc=70, xcount=10;
cardno=15;
il-12-i3-i4=0;
v1-v2-v3-v4-v5-v6=0;

popup15 = NRLX (Popup, console, XPos, 530, YPos, 20,
Label, "Extended Surface Target Parameters",
Width, 600,
Height, 800,
Background, "White",
FrameColor, "forest green",
NULL);

text68 = NRLX (TextEntry, popup15, XPos, xcount, YPos, ycount,
Label, "0 - Header, 1 - Detail",
Width, 580,
Height, 25,
Proc, stof1,
NULL);

text69 = NRLX (TextEntry, popup15, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "0-Point Model, 1-Area Model, # of point (if 11-1)",
Width, 580,
Height, 25,
Proc, stof2,
NULL);

text70 = NRLX (TextEntry, popup15, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Total # of detail points (detail cards must follow)",
Width, 580,
Height, 25,
Proc, stof3,
NULL);

text242 = NRLX (TextEntry, popup15, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Item 4",
Width, 580,
Height, 25,
Proc, stof4,
NULL);

text71 = NRLX (TextEntry, popup15, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Height (ft.)",
Width, 580,
Height, 25,
Proc, stof1,
NULL);

text72 = NRLX (TextEntry, popup15, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Area (sq. ft.)",
Width, 580,
Height, 25,
Proc, stof2,
NULL);

text243 = NRLX (TextEntry, popup15, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Var 3",
Width, 580,
Height, 25,
Proc, stof3,
NULL);

text244 = NRLX (TextEntry, popup15, XPos, xcount, YPos, (ycount-ycount+inc),

```

Wed Sep 21 16:37:05 1994

```

Label, "Var 4",
Width, 580,
Height, 25,
Proc, stof4,
NULL);

text245 = NRLx (TextEntry, popup15, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Var 5",
Width, 580,
Height, 25,
Proc, stof5,
NULL);

text246 = NRLx (TextEntry, popup15, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Var 6",
Width, 580,
Height, 25,
Proc, stof6,
NULL);

/* read in an existing file, make changes on card 15 */
if (cndt == 1) {
rewind (fpl);

fscanf (fpl, "%d%d%d%d%d%d%d%d", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);

while (cardno != 15) {
fseek (fpl, 1, 1);
fscanf (fpl, "%d%d%d%d%d%d%d%d", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
}

loc = ftell (fpl);
printf("win 15 1st loc = %d\n", loc);
loc = loc - 80;
printf("win 15 2nd loc = %d\n", loc);

printf (s11, "%d", i1);
printf (s12, "%d", i2);
printf (s13, "%d", i3);
printf (s14, "%d", i4);
printf (sv1, "%f", v1);
printf (sv2, "%f", v2);
printf (sv3, "%f", v3);
printf (sv4, "%f", v4);
printf (sv5, "%f", v5);
printf (sv6, "%f", v6);

NRLxLoadTextEntry (text68, s11);
NRLxLoadTextEntry (text69, s12);
NRLxLoadTextEntry (text70, s13);
NRLxLoadTextEntry (text242, s14);
NRLxLoadTextEntry (text71, sv1);
NRLxLoadTextEntry (text72, sv2);
NRLxLoadTextEntry (text243, sv3);
NRLxLoadTextEntry (text244, sv4);
NRLxLoadTextEntry (text245, sv5);
NRLxLoadTextEntry (text246, sv6);

Label, "Var 4",
Width, 580,
Height, 25,
Proc, stof4,
NULL);

}

/* Popup Window 16 - Parameters for Isometric Plot */
void isompt()
{
ycount=5, inc=65, xcount=5, ycount=290;
cardno=16;
i1=i2=i3=i4=0;
v1=v2=v3=v4=v5=v6=0;

popup16 = NRLx (Popup, console, XPos, 530, YPos, 20,
Width, 600,
Height, 800,
Background, "White",
FrameColor, "magenta",
NULL);

text73 = NRLx (Button, popup16, XPos, xcount, YPos, ycount,
Label, "Item 1: 0 - None (1,2,3)",
Proc, item16,
Data, 0,
Foreground, "salmon",
NULL);

text74 = NRLx (Button, popup16, XPos, 400, YPos, ycount,
Label, "1 - Signal/Noise",
Proc, item16,
Data, 1,
Foreground, "black",
NULL);

text75 = NRLx (Button, popup16, XPos, xcount, YPos, 30,
Label, "2 - Signal/(Clutter + Noise)",
Proc, item16,
Data, 2,
Foreground, "black",
NULL);

text76 = NRLx (Button, popup16, XPos, 300, YPos, 30,
Label, "3 - Probability of Detection",
Proc, item16,
Data, 3,
Foreground, "black",
NULL);

text77 = NRLx (Button, popup16, XPos, xcount, YPos, 55,
Label, "Height Independent Plots : 10 - Reset to zero(11,12,13)",
Proc, item16,
Data, 10,
Foreground, "black",
NULL);

text78 = NRLx (Button, popup16, XPos, xcount, YPos, 80,
Label, "11 - sh 1",
Proc, item16,
Data, 11,

```

Wed Sep 21 16:37:05 1994

```

Foreground, "black",
NULL);
text79 = NRLX (Button, popup16, XPos, 200, YPos, 80,
Label, "12 - sh 2",
Proc, item16,
Data, 12,
Foreground, "black",
NULL);

text80 = NRLX (Button, popup16, XPos, 400, YPos, 80,
Label, "13 - sh 3",
Proc, item16,
Data, 13,
Foreground, "black",
NULL);

text81 = NRLX (Button, popup16, XPos, xcount, YPos, 105,
Label, "Height Dependent Plots: 20 - Reset to zero(21,22,23)",
Proc, item16,
Data, 20,
Foreground, "black",
NULL);

text82 = NRLX (Button, popup16, XPos, xcount, YPos, 130,
Label, "21 - sh1",
Proc, item16,
Data, 21,
Foreground, "black",
NULL);

text83 = NRLX (Button, popup16, XPos, 200, YPos, 130,
Label, "22 - sh2",
Proc, item16,
Data, 22,
Foreground, "black",
NULL);

text84 = NRLX (Button, popup16, XPos, 400, YPos, 130,
Label, "23 - sh3",
Proc, item16,
Data, 23,
Foreground, "black",
NULL);

text85 = NRLX (Button, popup16, XPos, xcount, YPos, 160,
Label, "Item 2: 0 - Reset",
Proc, item2,
Data, 0,
Foreground, "salmon",
NULL);

text86 = NRLX (Button, popup16, XPos, 200, YPos, 160,
Label, "Height Independent Plots: 1 - Clutter/Noise",
Proc, item2,
Data, 1,
Foreground, "black",
NULL);

text87 = NRLX (Button, popup16, XPos, xcount, YPos, 185,
Label, "2 - Threshold Loss",
Proc, item2,
Data, 2,
Foreground, "black",
NULL);

text88 = NRLX (Button, popup16, XPos, 400, YPos, 185,
Label, "3 - Cancellation Ratio",
Proc, item2,
Data, 3,
Foreground, "black",
NULL);

text89 = NRLX (Button, popup16, XPos, xcount, YPos, 215,
Label, "Height Dependent Plots: 1 - Clutter/Noise",
Proc, item2,
Data, 1,
Foreground, "black",
NULL);

text90 = NRLX (Button, popup16, XPos, 400, YPos, 215,
Label, "2 - Signal/Noise",
Proc, item2,
Data, 2,
Foreground, "black",
NULL);

text91 = NRLX (Button, popup16, XPos, xcount, YPos, 240,
Label, "4 - Signal/(Clutter+Noise)",
Proc, item2,
Data, 4,
Foreground, "black",
NULL);

text92 = NRLX (Button, popup16, XPos, 300, YPos, 240,
Label, "8 - Prob. of Detection",
Proc, item2,
Data, 8,
Foreground, "black",
NULL);

text247 = NRLX (TextEntry, popup16, XPos, xcount, YPos, yycount,
Label, "Item 3",
Width, 550,
Height, 25,
Proc, sto13,
NULL);

text248 = NRLX (TextEntry, popup16, XPos, xcount, YPos, (yycount=yycount+inc),
Label, "Item 4",
Width, 550,
Height, 25,
Proc, sto14,
NULL);

text93 = NRLX (TextEntry, popup16, XPos, xcount, YPos, (yycount=yycount+inc),
Label, "Range Scale (nmi/in.)",
Width, 550,
Height, 25,
Proc, sto11,
NULL);

text94 = NRLX (TextEntry, popup16, XPos, xcount, YPos, (yycount=yycount+inc),

```

Wed Sep 21 16:37:05 1994

```

Label, "dB or Pd scale (dB/in. or unit/in.)",
Width, 50,
Height, 25,
Proc, stof2,
NULL);

text95 = NRLX (TextEntry, popup16, XPos, xcount, YPos, (yycount=yycount+inc),
Label, "Minimum dB or Pd (dB or unit)",
Width, 50,
Height, 25,
Proc, stof3,
NULL);

text96 = NRLX (TextEntry, popup16, XPos, xcount, YPos, (yycount=yycount+inc),
Label, "Maximum Z Dimension (in.)",
Width, 50,
Height, 25,
Proc, stof4,
NULL);

text97 = NRLX (TextEntry, popup16, XPos, xcount, YPos, (yycount=yycount+inc),
Label, "Height scale (ft/in.)",
Width, 50,
Height, 25,
Proc, stof5,
NULL);

text98 = NRLX (TextEntry, popup16, XPos, xcount, YPos, (yycount=yycount+inc),
Label, "Fixed XMAX for Plotting (nmi)",
Width, 50,
Height, 25,
Proc, stof6,
NULL);

/* read in an existing file, make changes on card 16 */
if (cndt == 1) {
    rewind (fp1);

    fscanf (fp1, "%d%d%d%d%d%f%f%f%f", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);

    while (cardno != 16) {
        fseek (fp1, 1, 1);
        fscanf (fp1, "%d%d%d%d%d%f%f%f%f", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
    }

    loc = ftell (fp1);
    printf("win 16 1st loc = %d\n", loc);
    loc = loc - 80;
    printf("win 16 2nd loc = %d\n", loc);

    printf (s11, "%d", i1);
    printf (s12, "%d", i2);
    printf (s13, "%d", i3);
    printf (s14, "%d", i4);
    printf (s15, "%f", v1);
    printf (s16, "%f", v2);
    printf (s17, "%f", v3);
    printf (s18, "%f", v4);
    printf (s19, "%f", v5);
    printf (s20, "%f", v6);

    NRLXLoadTextEntry (text247, s13);
    NRLXLoadTextEntry (text248, s14);
    NRLXLoadTextEntry (text93, s1);
    NRLXLoadTextEntry (text94, s2);
    NRLXLoadTextEntry (text95, s3);
    NRLXLoadTextEntry (text96, s4);
    NRLXLoadTextEntry (text97, s5);
    NRLXLoadTextEntry (text98, s6);
}

/* Popup Window 17 - Antenna Pattern Print/Plot Parameters */
void antpat()
{
    ycount=5, inc=70, xcount=10;
    cardno=17;
    i1=i2=i3=i4=0;
    v1=v2=v3=v4=v5=v6=0;

    popup17 = NRLX (Popup, console, XPos, 530, YPos, 10,
Label, "Antenna Pattern Print/Plot Parameters",
Width, 600,
Height, 800,
Background, "white",
FrameColor, "blue violet",
NULL);

    text100 = NRLX (TextEntry, popup17, XPos, xcount, YPos, ycount,
Label, "0-No Print or Plot, 1-Print, 2-Plot, 3-Print and Plot",
Width, 550,
Height, 25,
Proc, stof1,
NULL);

    text101 = NRLX (TextEntry, popup17, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "0 - Linear Scale, 1 - Polar Scale",
Width, 550,
Height, 25,
Proc, stof2,
NULL);

    text249 = NRLX (TextEntry, popup17, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Item 3",
Width, 550,
Height, 25,
Proc, stof3,
NULL);

    text250 = NRLX (TextEntry, popup17, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Item 4",
Width, 550,
Height, 25,
Proc, stof4,
NULL);

```



Wed Sep 21 16:37:05 1994

```

text102 = NRLX (TextEntry, popup17, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Minimum Phi (deg.)",
Width, 550,
Height, 25,
Proc, stof1,
NULL);

text103 = NRLX (TextEntry, popup17, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Maximum Phi (deg.)",
Width, 550,
Height, 25,
Proc, stof2,
NULL);

text104 = NRLX (TextEntry, popup17, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Delta Phi for Print & Plot (deg.)",
Width, 550,
Height, 25,
Proc, stof3,
NULL);

text105 = NRLX (TextEntry, popup17, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Scale Phi (deg./in.)",
Width, 550,
Height, 25,
Proc, stof4,
NULL);

text106 = NRLX (TextEntry, popup17, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Vertical Scale (dB/in.)",
Width, 550,
Height, 25,
Proc, stof5,
NULL);

text107 = NRLX (TextEntry, popup17, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Plot Height (in.)",
Width, 550,
Height, 25,
Proc, stof6,
NULL);

/* read in an existing file, make changes on card 17 */
if (cndt == 1) {
rewind (fpl);

fscanf (fpl, "%d%d%d%d%f%f%f%f", scardno, si1, si2, si3, si4, sv
1, sv2, sv3, sv4, sv5, sv6);

while (cardno != 17) {
fseek (fpl, 1, 1);
fscanf (fpl, "%d%d%d%d%f%f%f%f", scardno, si1, si2, si3, si4, sv
1, sv2, sv3, sv4, sv5, sv6);
}

loc = ftell (fpl);
printf("win 17 1st loc = %d\n", loc);
loc = loc - 80;
printf("win 17 2nd loc = %d\n", loc);

```

```

printf (si1, "%d", i1);
printf (si2, "%d", i2);
printf (si3, "%d", i3);
printf (si4, "%d", i4);
printf (sv1, "%f", v1);
printf (sv2, "%f", v2);
printf (sv3, "%f", v3);
printf (sv4, "%f", v4);
printf (sv5, "%f", v5);
printf (sv6, "%f", v6);

NRLXLoadTextEntry (text100, si1);
NRLXLoadTextEntry (text101, si2);
NRLXLoadTextEntry (text102, si3);
NRLXLoadTextEntry (text103, si4);
NRLXLoadTextEntry (text104, sv1);
NRLXLoadTextEntry (text105, sv2);
NRLXLoadTextEntry (text106, sv3);
NRLXLoadTextEntry (text107, sv4);
NRLXLoadTextEntry (text108, sv5);
NRLXLoadTextEntry (text109, sv6);
}

/* popup window 18 - Atmospheric Attenuation Parameters */
void atmattn()
{
ycount=5, inc=70, xcount=10;
cardno=18;
i1=i2=i3=i4=0;
v1=v2=v3=v4=v5=v6=0;

popup18 = NRLX (Popup, console, XPos, 530, YPos, 10,
Label, "Atmospheric Attenuation Parameters",
Width, 600,
Height, 800,
BackgroundColor, "white",
FrameColor, "lime green",
NULL);

text110 = NRLX (TextEntry, popup18, XPos, xcount, YPos, ycount,
Label, "0-No Attenuation, 1-Linear Attenuation, 2-Asymptotic atte
nuation",
Width, 550,
Height, 25,
Proc, stof1,
NULL);

text111 = NRLX (TextEntry, popup18, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "0 - Reset, # of Element",
Width, 550,
Height, 25,
Proc, stof2,
NULL);

text112 = NRLX (TextEntry, popup18, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "variable NPOLY",

```

Wed Sep 21 16:37:05 1994

```

width, 550,
Height, 25,
Proc, stoi3,
NULL);

text251 = NRLX (TextEntry, popup18, XPos, xcount, YPos, (Ycount=Ycount+inc),
Label, "Item 4",
Width, 550,
Height, 25,
Proc, stoi4,
NULL);

text113 = NRLX (TextEntry, popup18, XPos, xcount, YPos, (Ycount=Ycount+inc),
Label, "Attenuation value:(linear-dB/nmi) or (Asym.-dB/angle)",
Width, 550,
Height, 25,
Proc, stof1,
NULL);

text114 = NRLX (TextEntry, popup18, XPos, xcount, YPos, (Ycount=Ycount+inc),
Label, "variable A",
Width, 550,
Height, 25,
Proc, stof2,
NULL);

text115 = NRLX (TextEntry, popup18, XPos, xcount, YPos, (Ycount=Ycount+inc),
Label, "variable B",
Width, 550,
Height, 25,
Proc, stof3,
NULL);

text116 = NRLX (TextEntry, popup18, XPos, xcount, YPos, (Ycount=Ycount+inc),
Label, "variable R",
Width, 550,
Height, 25,
Proc, stof4,
NULL);

text117 = NRLX (TextEntry, popup18, XPos, xcount, YPos, (Ycount=Ycount+inc),
Label, "variable K",
Width, 550,
Height, 25,
Proc, stof5,
NULL);

text118 = NRLX (TextEntry, popup18, XPos, xcount, YPos, (Ycount=Ycount+inc),
Label, "variable 4",
Width, 550,
Height, 25,
Proc, stof6,
NULL);

/* read in an existing file, make changes on card 18 */
if (cndt == 1) {
    rewind (fp1);

    fscanf (fp1, "%d%d%d%d%d%f%f%f%f", scardno, si1, si2, si3, si4, sv
1, sv2, sv3, sv4, sv5, sv6);

    while (cardno != 18) {
        fseek (fp1, 1, 1);
        fscanf (fp1, "%d%d%d%d%d%f%f%f%f", scardno, si1, si2, si3, si4, sv
1, sv2, sv3, sv4, sv5, sv6);
    }

    loc = ftell (fp1);
    printf("win 18 1st loc = %d\n", loc);
    loc = loc - 80;
    printf("win 18 2nd loc=%d\n", loc);

    sprintf (si1, "%d", il);
    sprintf (si2, "%d", i2);
    sprintf (si3, "%d", i3);
    sprintf (si4, "%d", i4);
    sprintf (sv1, "%f", v1);
    sprintf (sv2, "%f", v2);
    sprintf (sv3, "%f", v3);
    sprintf (sv4, "%f", v4);
    sprintf (sv5, "%f", v5);
    sprintf (sv6, "%f", v6);

    NRLXLoadTextEntry (text110, si1);
    NRLXLoadTextEntry (text111, si2);
    NRLXLoadTextEntry (text112, si3);
    NRLXLoadTextEntry (text251, si4);
    NRLXLoadTextEntry (text113, sv1);
    NRLXLoadTextEntry (text114, sv2);
    NRLXLoadTextEntry (text115, sv3);
    NRLXLoadTextEntry (text116, sv4);
    NRLXLoadTextEntry (text117, sv5);
    NRLXLoadTextEntry (text118, sv6);
}

/* Popup Window 19 - Rain Attenuation Parameters */

void rainatt()
{
    Ycount=5, inc=70, xcount=10;
    cardno=19;
    il=i2=i3=i4=0;
    v1=v2=v3=v4=v5=v6=0;

    popup19 = NRLX (Popup, console, XPos, 530, YPos, 10,
Label, "Rain Attenuation Parameters",
Width, 600,
Height, 800,
Background, "white",
FrameColor, "orange",
NULL);

    text120 = NRLX (TextEntry, popup19, XPos, xcount, YPos, Ycount,
Label, "0 - None, 1 - Rain",
Width, 550,
Height, 25,
Proc, stoi1,

```

Wed Sep 21 16:37:05 1994

```

        NULL);
text252 = NRLX (TextEntry, popup19, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Item 2",
Width, 550,
Height, 25,
Proc, sto12,
NULL);

text253 = NRLX (TextEntry, popup19, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Item 3",
Width, 550,
Height, 25,
Proc, sto13,
NULL);

text254 = NRLX (TextEntry, popup19, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Item 4",
Width, 550,
Height, 25,
Proc, sto14,
NULL);

text121 = NRLX (TextEntry, popup19, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Attenuation (dB/nmi)",
Width, 550,
Height, 25,
Proc, sto11,
NULL);

text122 = NRLX (TextEntry, popup19, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Minimum Range (nmi)",
Width, 550,
Height, 25,
Proc, sto12,
NULL);

text123 = NRLX (TextEntry, popup19, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Maximum Range (nmi)",
Width, 550,
Height, 25,
Proc, sto13,
NULL);

text255 = NRLX (TextEntry, popup19, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "var 4",
Width, 550,
Height, 25,
Proc, sto14,
NULL);

text256 = NRLX (TextEntry, popup19, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "var 5",
Width, 550,
Height, 25,
Proc, sto15,
NULL);

text257 = NRLX (TextEntry, popup19, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "var 6",
Width, 550,
Height, 25,
Proc, sto16,
NULL);

/* popup window 20 - Jammer Parameters */
void jammer()
{
    ycount=5, inc=70, xcount=10;
    cardno=20;
    il=i2=i3=i4=0;
    vl=v2=v3=v4=v5=v6=0;

    popup20 = NRLX (Popup, console, XPos, 530, YPos, 10,
Label, "Jammer Parameters",
Width, 600,
Height, 800,
Background, "white",
FrameColor, "yellow",
NULL);

    text258 = NRLX (TextEntry, popup20, XPos, xcount, YPos, ycount,
Label, "Item 1",
Width, 550,
Height, 25,
Proc, sto11,
NULL);

    text259 = NRLX (TextEntry, popup20, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Item 2",
Width, 550,
Height, 25,
Proc, sto12,
NULL);

    text260 = NRLX (TextEntry, popup20, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Item 3",
Width, 550,
Height, 25,
Proc, sto13,
NULL);

    text261 = NRLX (TextEntry, popup20, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Item 4",
Width, 550,
Height, 25,
Proc, sto14,
NULL);

    text130 = NRLX (TextEntry, popup20, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Jammer Effective Radiated Power (W)",
Width, 550,

```

Wed Sep 21 16:37:05 1994

```

Height, 25,
Proc, stof1,
NULL);

text131 = NRLX (TextEntry, popup20, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Jammer Bandwidth (MHz)",
Width, 550,
Height, 25,
Proc, stof2,
NULL);

text132 = NRLX (TextEntry, popup20, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Jammer Range (nmi)",
Width, 550,
Height, 25,
Proc, stof3,
NULL);

text133 = NRLX (TextEntry, popup20, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Antenna Sidelobe Gain (dB)",
Width, 550,
Height, 25,
Proc, stof4,
NULL);

text134 = NRLX (TextEntry, popup20, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "SLC Cancellation Ratio (dB)",
Width, 550,
Height, 25,
Proc, stof5,
NULL);

text262 = NRLX (TextEntry, popup20, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Var 6",
Width, 550,
Height, 25,
Proc, stof6,
NULL);

/* read in an existing file, make changes on card 20 */

if (cndt == 1) {
    rewind (fp1);

    fscanf (fp1, "%d%d%d%d%d%f%f", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);

    while (cardno != 20) {
        fseek (fp1, 1, 1);
        fscanf (fp1, "%d%d%d%d%d%f%f", &cardno, &i1, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
    }

    loc = ftell (fp1);
    printf("win 20 1st loc = %d\n", loc);
    loc = loc - 80;
    printf("win 20 2nd loc=%d\n", loc);

    sprintf (s1, "%d", i1);
    sprintf (s2, "%d", i2);

```

```

    sprintf (s13, "%d", i3);
    sprintf (s14, "%d", i4);
    sprintf (sv1, "%f", v1);
    sprintf (sv2, "%f", v2);
    sprintf (sv3, "%f", v3);
    sprintf (sv4, "%f", v4);
    sprintf (sv5, "%f", v5);
    sprintf (sv6, "%f", v6);

    NRLXLoadTextEntry (text258, s11);
    NRLXLoadTextEntry (text259, s12);
    NRLXLoadTextEntry (text260, s13);
    NRLXLoadTextEntry (text261, s14);
    NRLXLoadTextEntry (text130, sv1);
    NRLXLoadTextEntry (text131, sv2);
    NRLXLoadTextEntry (text132, sv3);
    NRLXLoadTextEntry (text133, sv4);
    NRLXLoadTextEntry (text134, sv5);
    NRLXLoadTextEntry (text262, sv6);
}

/* popup window 1 - Execute */

void execute()
{
    ycount=5, inc=70, xcount=10;
    cardno=1;
    i1=i2=i3=i4=0;
    v1=v2=v3=v4=v5=v6=0;

    popup1 = NRLX (Popup, console, XPos, 530, YPos, 10,
        Label, "Execute",
        Width, 600,
        Height, 800,
        Background, "white",
        FrameColor, "dark turquoise",
        NULL);

    text140 = NRLX (TextEntry, popup1, XPos, xcount, YPos, ycount,
        Label, "0 - Clutter + Noise, 1 - Noise only",
        Width, 550,
        Height, 25,
        Proc, stoil,
        NULL);

    text141 = NRLX (TextEntry, popup1, XPos, xcount, YPos, (ycount=ycount+inc),
        Label, "0 - No pattern, print, plot, 1 - print, plot, pattern, continue",
        Width, 550,
        Height, 25,
        Proc, stoil2,
        NULL);

    text142 = NRLX (TextEntry, popup1, XPos, xcount, YPos, (ycount=ycount+inc),
        Label, "0 - Print, 1 - No Print",
        Width, 550,
        Height, 25,
        Proc, stoil3,
        NULL);

```

Wed Sep 21 16:37:05 1994

```

text263 = NRLx (TextEntry, popup1, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Item 4",
Width, 550,
Height, 25,
Proc, stof4,
NULL);

text143 = NRLx (TextEntry, popup1, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Single Target Altitude (ft.)",
Width, 550,
Height, 25,
Proc, stof1,
NULL);

text264 = NRLx (TextEntry, popup1, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Var 2",
Width, 550,
Height, 25,
Proc, stof2,
NULL);

text265 = NRLx (TextEntry, popup1, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Var 3",
Width, 550,
Height, 25,
Proc, stof3,
NULL);

text266 = NRLx (TextEntry, popup1, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Var 4",
Width, 550,
Height, 25,
Proc, stof4,
NULL);

text267 = NRLx (TextEntry, popup1, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Var 5",
Width, 550,
Height, 25,
Proc, stof5,
NULL);

text268 = NRLx (TextEntry, popup1, XPos, xcount, YPos, (ycount=ycount+inc),
Label, "Var 6",
Width, 550,
Height, 25,
Proc, stof6,
NULL);

/* read in an existing file, make changes on card 1 */
if (cndt == 1) {
    rewind (fp1);
    fscanf (fp1, "%d%d%d%d%d%f%f%f", scardno, si1, si2, si3, si4, sv
1, sv2, sv3, sv4, sv5, sv6);
    while (cardno != 1) {
        fseek (fp1, 1, 1);
        fscanf (fp1, "%d%d%d%d%d%f%f%f", scardno, si1, si2, si3, si4, sv
1, sv2, sv3, sv4, sv5, sv6);
    }

    loc = ftell (fp1);
    printf ("win 1 1st loc = %d\n", loc);
    loc = loc - 80;
    printf ("win 1 2nd loc=%d\n", loc);

    sprintf (si1, "%d", il);
    sprintf (si2, "%d", i2);
    sprintf (si3, "%d", i3);
    sprintf (si4, "%d", i4);
    sprintf (sv1, "%f", v1);
    sprintf (sv2, "%f", v2);
    sprintf (sv3, "%f", v3);
    sprintf (sv4, "%f", v4);
    sprintf (sv5, "%f", v5);
    sprintf (sv6, "%f", v6);

    NRLxLoadTextEntry (text140, si1);
    NRLxLoadTextEntry (text141, si2);
    NRLxLoadTextEntry (text142, si3);
    NRLxLoadTextEntry (text263, si4);
    NRLxLoadTextEntry (text143, sv1);
    NRLxLoadTextEntry (text264, sv2);
    NRLxLoadTextEntry (text265, sv3);
    NRLxLoadTextEntry (text266, sv4);
    NRLxLoadTextEntry (text267, sv5);
    NRLxLoadTextEntry (text268, sv6);
}

/* Popup Window 11 - Title for printout and plot */
void title()
{
    cardno=11;
    ycount=5, xcount=10, inc=70;
    il=i2-i3=i4=0;
    v1=v2=v3=v4=v5=v6=0;

    popup11 = NRLx (Popup, console, XPos, 530, YPos, 20,
Label, "Title for Printout and Plot",
Width, 600,
Height, 800,
Background, "white",
FrameColor, "dark green",
NULL);

    text150 = NRLx (TextEntry, popup11, XPos, xcount, YPos, ycount,
Label, "# of Title Line to follow (maximum is 4)",
Width, 580,
Height, 25,
Proc, stof1,
NULL);
}

```

Wed Sep 21 16:37:05 1994

```

text151 = NRLX (TextEntry, popup11, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Line one",
Width, 580,
Height, 25,
Proc, string1,
NULL);

text152 = NRLX (TextEntry, popup11, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Line two",
Width, 580,
Height, 25,
Proc, string2,
NULL);

text153 = NRLX (TextEntry, popup11, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Line three",
Width, 580,
Height, 25,
Proc, string3,
NULL);

text154 = NRLX (TextEntry, popup11, XPos, xcount, YPos, (ycount-ycount+inc),
Label, "Line four",
Width, 580,
Height, 25,
Proc, string4,
NULL);

/* read in an existing file, read a line and check for cardno 11 */
if (cndt == 1) {
    rewind (fpl);
    fscanf (fpl, "%d%d%d%d%d%d%d%d%d%d", &cardno, &il, &i2, &i3, &i4, &v1, &v2, &v
3, &v4, &v5, &v6);
    while (cardno != 11) {
        fseek (fpl, 1, 1);
        fscanf (fpl, "%d%d%d%d%d%d%d%d%d%d", &cardno, &il, &i2, &i3, &i4, &v
1, &v2, &v3, &v4, &v5, &v6);
    }
    loc = ftell (fpl);
    printf(" card 11 - read in loc = %d\n", loc);
    sprintf (sil, "%d", il);
    NRLXLoadTextEntry (text150, sil);
    loc2=loc+3;
    printf(" card 11 - read in loc2 = %d\n", loc2);
    if (il == 1) {
        fseek (fpl, loc2, 0);
        fgets (rstr1, 78, fpl);
        NRLXLoadTextEntry (text151, rstr1);
    }
    else if (il == 2) {
        fseek (fpl, loc2, 0);
        fgets (rstr1, 78, fpl);
        fseek (fpl, loc2+81, 0);
        fgets (rstr2, 78, fpl);
    }
    else if (il == 3) {
        fseek (fpl, loc2, 0);
        fgets (rstr1, 78, fpl);
        fseek (fpl, loc2+81, 0);
        fgets (rstr2, 78, fpl);
        fseek (fpl, loc2+162, 0);
        fgets (rstr3, 78, fpl);
        NRLXLoadTextEntry (text151, rstr1);
        NRLXLoadTextEntry (text152, rstr2);
        NRLXLoadTextEntry (text153, rstr3);
    }
    else if (il == 4) {
        fseek (fpl, loc2, 0);
        fgets (rstr1, 78, fpl);
        fseek (fpl, loc2+81, 0);
        fgets (rstr2, 78, fpl);
        fseek (fpl, loc2+162, 0);
        fgets (rstr3, 78, fpl);
        fseek (fpl, loc2+243, 0);
        fgets (rstr4, 78, fpl);
        NRLXLoadTextEntry (text151, rstr1);
        NRLXLoadTextEntry (text152, rstr2);
        NRLXLoadTextEntry (text153, rstr3);
        NRLXLoadTextEntry (text154, rstr4);
    }
}

/* title for printout and plot */
void string1 (String ln1)
{
    /* printf("string1 - str1=%d\n", str1); */
    /* printf("C - line one : %s\n", ln1); */
    printf(str1, "%s", ln1);
    printf ("string1 - str1= %s\n", str1);
    printf (astr1, "%-78s", str1);
    if (cndt == 1) line1=1;
}

void string2 (String ln2)
{
    printf(str2, "%s", ln2);
    printf ("string2 - str2= %s\n", str2);
    printf (astr2, "%-78s", str2);
    if (cndt == 1) line2=1;
}

```

Wed Sep 21 16:37:05 1994

```

void string3 (String ln3)
{
    sprintf(str3, "%s", ln3);
    printf("string3 - str3=%s\n", str3);
    sprintf (astr3, "%-78s", str3);
    if (cndt == 1) line3=1;
}

void string4 (String ln4)
{
    sprintf(str4, "%s", ln4);
    printf("string4 - str4=%s\n", str4);
    sprintf (astr4, "%-78s", str4);
    if (cndt == 1) line4=1;
}

/* item 1 on card 16 */
void item16 (long data)
{
    printf("item 1 on card 16: %d\n", data);
    il=data;
}

/* item 2 on card 16 */
void item2 (long data)
{
    printf("item 2 on card 16: %d\n", data);
    i2=data;
}

/* string to integer */
void stoi1 (String var1)
{
    printf("Incoming from text widget: %s\n", var1);
    sscanf (var1, "%d", &i1);
}

/* string to floating point */
void stof1 (String var5)
{
    printf("Incoming from text widget: %s\n", var5);
    /* MUST use ADDRESS of numeric variable---- &v1 -----*/
    sscanf (var5, "%f", &v1);
    printf("Result of conversion: %f\n", v1);
}

void stof2 (String var6)
{
    sscanf (var6, "%f", &v2);
}

void stof3 (String var7)
{
    sscanf (var7, "%f", &v3);
}

void stof4 (String var8)
{
    sscanf (var8, "%f", &v4);
}

void stof5 (String var9)
{
    sscanf (var9, "%f", &v5);
}

void stof6 (String var10)
{
    sscanf (var10, "%f", &v6);
}

/* Open a new data file */
void openfile()
{
    cndt=2;
}

/* get the filename */
printf (" File name = ");
gets (fname);
if ((fp = fopen(fname, "w")) == NULL)
{
    printf("Unable to open %s\n", fname);
    exit(1);
}

```

```

    }

    /* Close the data file */
    void closeinfile()
    {
        cardno=0;
        il=i2=i3=i4=0;
        v1=v2=v3=v4=v5=v6=0;

        fprintf(fp,"%2d%4d%6d%4d%10.4f%10.4f%10.4f%10.4f%10.4f\n", ca
rdno,i1,i2,i3,i4,v1,v2,v3,v4,v5,v6);

        fclose(fp);
    }

    /* save the window */
    void saveinf()
    {
        if (cardno == 11) {
            loc1=loc-80;
            fseek (fpl, loc1, 0);
            printf ("% saveinf-loc= %d\n", loc1);

            fprintf (fpl,"%2d%4d%6d%4d%10.4f%10.4f%10.4f%10.4f%10.4f\n",
cardno,i1,i2,i3,i4,v1,v2,v3,v4,v5,v6);

            printf ("% saveinf-line1= %d\n", line1);
            printf ("% saveinf-line2= %d\n", line2);
            printf ("% saveinf-line3= %d\n", line3);
            printf ("% saveinf-line4= %d\n", line4);
            printf ("% saveinf-loc2 = %d\n", loc2);

            fseek (fpl, loc2, 0);

            if (il == 1 && line1 == 1) fputs (astr1, fpl);

            else if (il == 2 && line1 == 1) fputs (astr1, fpl);
            else if (il == 2 && line2 == 1) {
                fseek (fpl, loc2+81, 0);
                fputs (astr2, fpl);
            }

            else if (il == 3 && line1 == 1) fputs (astr1, fpl);
            else if (il == 3 && line2 == 1) {
                fseek (fpl, loc2+162, 0);
                fputs (astr3, fpl);
            }

            else if (il == 4 && line1 == 1) fputs (astr1, fpl);
            else if (il == 4 && line2 == 1) {
                fseek (fpl, loc2+81, 0);
                fputs (astr2, fpl);
            }

            else if (il == 4 && line3 == 1) {
                fseek (fpl, loc2+162, 0);
                fputs (astr3, fpl);
            }

            else if (il == 4 && line4 == 1) {
                fseek (fpl, loc2+243, 0);
            }
        }

        printf ("% File name for read = ");
        gets (infname);
        if ((fpl = fopen(infname, "r+")) == NULL)
        {
            printf("Unable to open %s\n", infname);
            exit(1);
        }

        /* open an existing data file */
        void openinfile()
        {
            cndt=1;

            printf ("% File name for read = ");
            gets (infname);
            if ((fpl = fopen(infname, "r+")) == NULL)
            {
                printf("Unable to open %s\n", infname);
                exit(1);
            }
        }

        fprintf (fp,"%2d%4d%6d%4d%10.4f%10.4f%10.4f%10.4f%10.4f\n", c
ardno,i1,i2,i3,i4,v1,v2,v3,v4,v5,v6);

        if (il == 1) fprintf (fp, "%2d%-78s\n", cardno, astr1);
        else if (il == 2) fprintf (fp, "%2d%-78s\n%2d%-78s\n", cardno, astr1,
cardno, astr2);

        else if (il == 3) fprintf (fp, "%2d%-78s\n%2d%-78s\n%2d%-78s\n", cardn
o, astr1, cardno, astr2, cardno, astr3);
        else if (il==4) fprintf(fp,"%2d%-78s\n%2d%-78s\n%2d%-78s\n%2d%-78s\n",
cardno, astr1, cardno, astr2, cardno, astr3, cardno, astr4);
    }

    else
        fprintf (fp,"%2d%4d%6d%4d%10.4f%10.4f%10.4f%10.4f%10.4f\n", c
ardno,i1,i2,i3,i4,v1,v2,v3,v4,v5,v6);

    /* open an existing data file */
    void openinfile()
    {
        cndt=1;

        printf ("% File name for read = ");
        gets (infname);
        if ((fpl = fopen(infname, "r+")) == NULL)
        {
            printf("Unable to open %s\n", infname);
            exit(1);
        }
    }

```



Wed Sep 21 16:37:05 1994

```
fputs (astr4, fpl);
}
}
else if (cardno != 11) {
    fseek (fpl, loc, 0);
    fprintf (fpl, "%12d%4d%6d%4d%10.4f%10.4f%10.4f%10.4f%10.4f%10.4f\n",
cardno, i1, i2, i3, i4, v1, v2, v3, v4, v5, v6);
}
}

/* Quit function */
void quit()
{
    exit(0);
}
```